AUTOMATED LIQUOR DISPENSER USING PIC16F877 WITH LIQUID LEVEL SENSOR

by

Jeffrey R. Cutaran
Carla Marie M. Fernandez
John Patrick R. Galang

A Design Report Submitted to the School of Electrical Engineering, Electronics Engineering, and Computer Engineering in Partial Fulfilment of the Requirements for the Degree

Bachelor of Science in Computer Engineering

Mapua Institute of Technology
September 2011

Approval Sheet

Mapua Institute of Technology
School of EECE

This is to certify that I have supervised the preparation of and read the design report prepared by **Jeffrey R. Cutaran, Carla Marie M. Fernandez,** and **John Patrick R. Galang** entitled **AUTOMATED LIQUOR DISPENSER USING PIC16F877 WITH LIQUID LEVEL SENSOR** and that the said report has been submitted for final examination by the Oral Examination Committee.

Jesus M. Martinez
Design Adivser

As members of the Oral Examination Committee, we certify that we have examined this design report, presented before the committee on **July 6, 2011**, and hereby recommended that it be accepted in partial fulfilment of the requirements for the degree in **Bachelor of Science in Computer Engineering.**

Dionis Padilla
Panel Member

Ayra Panganiban
Panel Member

Joshua Cuesta
Panel Member

This design report is hereby approved and accepted by the School of Electrical Engineering, Electronics Engineering, and Computer Engineering in partial fulfilment of the requirements for the degree in **Bachelor of Science in Computer Engineering.**

Felicito S. Caluyo, Ph. D.
Dean, School of EECE

Acknowledgement

We would like to extend our heartfelt gratitude to those who offered their help and support which brought this design report to completion.

First and foremost, to the Almighty God for guiding and giving us wisdom and for His continuous blessings;

To our friends and families, for their endless support in any form, for their prayers and advises for the success of this design;

To our Adviser, Engr. J.M. Martinez, for helping us out in the design.

Lastly, to our professor, Engr. Noel Linsangan, for his patience, concern, knowledge and expertise he had imparted to us.

Jeffrey R. Cutaran

Carla Marie M. Fernandez

John Patrick R. Galang

ROLES AND RESPONSIBILITIES

The design project is a concerted effort of each member of the group. They combined all of the gathered information, analyzed them, took what is important and useful and from there they created the design. For the development of the prototype, each of the members also did their part.

The following shows the detailed list of roles and responsibilities for each member:

Jeffrey Cutaran	• Compilation of documentation • Buying of parts • Designing of power supply circuit • Testing of prototype • Programming of microcontroller
Carla Fernandez	• Chapter 1, 2, 3, and 5 • Testing of prototype • Designing of liquid level circuit • Buying of parts
John Patrick Galang	• Chapter 1, 3, 4, and 5 • Designing of microcontroller circuit • Mounting and soldering parts on PCB • Testing of prototype • Programming of microcontroller

Table 1 Roles and Responsibilities

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

ABSTRACT

The design is about the implementation of an automated microcontroller-based liquor dispenser that can automatically provide mixed drinks. The machine uses programmed microcontrollers that function as the brain of the system. It has predefined programs and instructions that are responsible for the liquor dispensing processes that the machine will perform as directed by the user. The quantity of each ingredient to be dispensed is controlled by a microcontroller. After which, the liquor is dispensed in a glass through a hose. The ingredient level for each container is monitored by a liquid level sensor located inside the container. The system offers a choice of 15 pre-programmed mixes and based ingredients. The design can help bartenders in making drinks.

Keywords: Automated liquor dispenser, microcontroller-based provider, liquor dispenser.

Chapter 1

DESIGN BACKGROUND AND INTRODUCTION

The introduction gives a general overview of the design project, giving the reader the background or basis of the problem to be reported.

Background

A typical bar scenario during happy hour involves people ordering drinks simultaneously, with only a few bartenders to tend to them. As a result, the quality of drinks produced may become inconsistent.

The design is aimed at solving that problem. Having an automatic liquor dispenser will lessen manpower needed behind bar counters. It will also reduce the amount of time needed in making drinks and will improve accuracy in terms of the liquors' volume because it dispenses the same amount with minimal error.

The design is composed of five containers which store the five basic liquors. It uses a microcontroller that serves as the brain of the system to which all the operating processes of each component are chronologically programmed in it. It is programmed to produce liquor that the user will input through the use of a keypad. The design also uses a pump that is controlled by relays which are used to transfer liquor to the dispenser through hose pipes. The microcontroller is programmed to control and monitor the quantity

of liquor to be dispensed, and it controls the LCD which displays the user's selection. The liquor level for each of the five containers is monitored by a water level sensor located inside the container. Ten LEDs indicate whether the liquor level is at a low level or close to empty, and it will alert the microcontroller and will no longer allow the dispenser to dispense once the liquor level is close to empty. This will be displayed in the LCD as well. The combined features make the design suitable for use inside bars.

Statement of the Problem

A bartender mixing drinks is prone to human error. An example is when ingredients mixed together may not be the right one. Another is the amount of alcohol may vary causing inconsistencies. Also numerous orders prolong the wait time of customers to get their drinks. Lastly, bartenders are prone to exhaustion. These errors cause bar owners to lose their customers. As a solution, the group decided to design a machine that will reduce if not eliminate the said errors.

Objective of the Design

The general objective pertains to the main purpose of the design which addresses the statement of the problem. The specific objective identifies the functionalities and features that the device should possess.

General Objectives

a. To be able to build an automated liquor dispenser.
b. To be able to use the least possible cost in designing the project.

Specific Objectives

a. To create a design that is capable of dispensing close to the exact amount of ingredients for all drinks programmed in it.
b. To be able to produce numerous drinks, provided that the container is still not empty, without altering quality.
c. To prevent foreign matter from mixing with the drinks.

Significance and Impact of the Design

The proposed design is intended for bar owners who serve drinks and other alcoholic beverages. This design could help reduce the work of bartenders by being the one that pours drinks into a glass. Also, it is very timely because technology today is rapidly changing and so keeping pace with the demand in the bar industry is always necessary.

The impact of the design on the economy is that it will contribute mainly to the bar industry. The design will lessen manual labor needed behind bars. It controls beverage pour sizes and provides portioned drinks. The design also prevents product loss by eliminating over-pouring and spillage of liquor as well as breakages due to bottle mishandling. The design can contribute to the full utilization of a bar's resources.

Scope and Delimitation

The design emulates the functionality (alcohol drink production) of a bartender. It is capable of the following:

Scope

1. Dispensing alcoholic beverages of 10 different mixes and five basics.
2. Controlling the amount of ingredients used in making the drink.
3. Produces liquor close to the exact amount of drinks for the required ingredients.
4. Device is powered by a 220-VAC source.
5. The design has sensors that would detect if the liquor level is below low level or close to empty.

Delimitation

1. The automated liquor dispenser cannot function as a vending machine. Thereby, it does not fully eliminate the use of manual labour.
2. The design cannot function as a mixer.
3. The design can only produce drinks that are programmed into it. It can not generate new types of drinks.
4. The amount of liquid dispensed is not exact compared to what it is supposed to dispense due to the residue from previous dispensing.

Definition of Terms

1. Automate - to convert to automatic operation.

2. Block Diagram - a diagram of a system, in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks.
3. Capacitor - a passive electronic component consisting of a pair of conductors separated by an adielectric (insulator). When there is a potential difference (in voltage) across the conductors, a static electric field develops in the dielectric that stores energy and produces a mechanical force between the conductors.
4. Circuit - a simplified conventional graphical representation of an electrical circuit.
5. Crystal Oscillator - an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency.
6. Density - defined as mass per unit volume. The symbol most often used for density is ρ (the Greek letter rho). In some cases, density is also defined as weight per unit volume although this quantity is more properly called specific weight. Different materials usually have different densities, so density is an important concept regarding buoyancy, purity and packaging.
7. Diode – a two-terminal electronic device that permits current flow predominantly in only one direction. A diode has a low resistance to

electric current in one direction and a high resistance to it in the reverse direction.

8. Dispenser - container so designed that the contents can be used in prescribed amounts.
9. Embedded Systems - computer systems that cannot be programmed by the user because they are pre-programmed for a specific task and are buried within the equipment they serve.
10. Foreign Matter - most commonly refers to the presence of unwanted or undesirable material present in foods or chemicals.
11. Hardware - a general term for the physical artifacts of a technology.
12. Impedance - describes a measure of opposition to alternating current (AC). Electrical impedance extends the concept of resistance to AC circuits, describing not only the relative amplitudes of the voltage and current, but also the relative phases. When the circuit is driven with direct current (DC) there is no distinction between impedance and resistance; the latter can be thought of as impedance with a zero phase angle.
13. Keypad - a set of buttons arranged in a block or "pad" which usually bear digits.
14. LCD - digital display that uses liquid crystal cells that change reflectivity in an applied electric field; used for portable computer displays, watches, etc.

15. LED (light emitting diode) – a diode that produces visible or infrared light when subjected to an electric current as a result of electroluminescence.
16. Level Sensors - detect the level of substances that flow, including liquids, slurries, granular materials, and powders. Fluids and fluidized solids flow to become essentially level in their containers (or other physical boundaries) because of gravity whereas most bulk solids pile at an angle of repose to a peak.
17. Microcontroller - a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.
18. Ohm's Law - states that the current through a conductor between two points is directly proportional to the potential difference or voltage across the two points, and inversely proportional to the resistance between them
19. PCB (printed circuit board) - used to mechanically support and electrically connect electronic components using conductive pathways, tracks or signal traces; etched from copper sheets laminated onto a non-conductive substrate.
20. PIC16F877A - a small piece of semiconductor integrated circuit. The package type of this integrated circuit is DIP, which stands for Dual Inline Package for semiconductor IC. This package is very easy to be soldered onto a stripboard. However using a DIP socket is much easier so that this chip can be plugged and removed from the development board.

21. Power Supply - a device that supplies electrical energy to one or more electric loads.
22. Programming - the process of designing, writing, testing, debugging / troubleshooting, and maintaining the source code of computer programs.
23. Prototype - an original, full-scale, and usually working model of a new product or new version of an existing product.
24. Pump - a mechanical device that moves fluid or gas by pressure or suction
25. Relay – an electrical device such that current flowing through it in one circuit can switch on and off a current in a second circuit
26. Resistor – a two-terminal electric circuit component that offers opposition to an electric current. Resistors are normally designed and operated so that, with varying levels of current, variations of their resistance values are negligible.
27. Schematic Diagram - represents the elements of a system using abstract, graphic symbols rather than realistic pictures.
28. Sensor - any device that receives a signal or stimulus (as heat or pressure or light or motion etc.) and responds to it in a distinctive manner.
29. Software - the collection of computer programs and related data that provide instructions that tell a computer what to do.

30. Soldering - a process in which two or more metal items are joined together by melting and flowing a filler metal into the joint, the filler metal having a relatively low melting point.
31. System Flowchart - a graphical representation of a process, such as a manufacturing operation or a computer operation, indicating the various steps taken as the product moves along the production line or the problem moves through the computer.
32. Transient Response - the response of a system to a change from equilibrium. A simple example would be the output of a 5 volt DC power supply when it is turned on: the transient response is from the time the switch is flipped until the output reaches a steady 5 volts. At this time the power supply reaches its steady-state response of a constant 5 volts.
33. Transistor - a semiconductor device used to amplify and switch electronic signals. It is made of a solid piece of semiconductor material, with at least three terminals for connection to an external circuit. A voltage or current applied to one pair of the transistor's terminals changes the current flowing through another pair of terminals.

Chapter 2

REVIEW OF RELATED LITERATURE

The researchers used the following journal articles and other sources as references and guide in developing the design, an Automated Liquor Dispenser.

The design is a device that dispenses liquor. It uses a PIC16F877A as its microprocessor which is responsible for managing the different processes the design does.

An article about the intelligent water dispenser system based on embedded systems by Jinhuang (2003) discusses a device that uses a single-bus temperature sensor DS18B20 to measure real-time temperature of drinking fountains, provides a calendar and time through clock chip DS1302, receives information from a remote through HS0038B, and displays the calendar and time as well as the current temperature value through LCD12864. The system was designed based on microcontroller STC89C52. Although the design does not incorporate the said function, the researchers studied the system's overall design concept, the hardware circuit and software flow chart and design, and the use of the fitting algorithm. The system has such functions as remote control, temperature control, cooling, variable power heating, and it has a high level of safety, stability, intelligent control and low power consumption.

Another article by Garvie (2002) describes a liquid tot dispenser. The liquid tot dispenser has a container for the liquid and a thimble that has an inlet from the container and an outlet, both being valve controlled. The interior of the thimble has a vent to atmosphere and a second vent from a low level in the container to atmosphere. The article explained how to control the amount of liquid can be controlled once it is flowing.

The article of Daniel N. Campau of Grand Rapids, MI, presents a flow control device for providing variable resistance to liquid flow through a flow passageway. A cylindrical housing communicates with the passageway. The housing has a sidewall, and an inlet and an outlet each disposed at two ends. A vortex generator is located within the housing, and has a base spaced from the inlet end of the housing and an annular flow guide radially spaced from the housing sidewall. The flow guide includes a number of slots. Liquid enters the housing through the inlet and is directed outside the vortex generator and through the slots. This creates a vortex flow path within the generator as the liquid flows to the housing outlet, so that as the pressure of the liquid at the inlet increases the flow factor of the device decreases to reduce the liquid flow rate through the device at higher inlet pressures.

The article of Bartoletti Sr (1999) explain the concept of a beverage dispenser that includes an outer housing having a water bath tank therein and a refrigeration retaining component area therein positioned directly adjacent and next to the water bath tank. A refrigeration chassis provides for retention and

carrying of a refrigeration system including a compressor, a condenser and powered cooling fan and an evaporator. The chassis and refrigeration components form a U-shape wherein one "leg" thereof consists of a rectangular sheet metal frame for retaining the compressor and condenser and the other leg consists of the evaporator. The bridge or end portion of the U-shape consists of a horizontal top plate portion of the chassis and the fluid connection between the evaporator and the condenser. The evaporator is suspended from the horizontal top plate. The U-shape of the chassis and refrigeration components facilitates a method of manufacture. In particular, a carbonator, syrup cooling coils and a water cooling coil are first positioned in the water bath tank at an end thereof adjacent the end of the housing on which a plurality of beverage dispensing valves are secured. The assembled chassis with refrigeration components secured thereto is then lifted and lowered into the dispenser housing wherein the evaporator is placed into the water bath tank along an end thereof opposite from the carbonator and cooling coils, and where the compressor and condenser are placed into the refrigeration component retaining area.

Alongside the beverage dispenser is a study by Peckels (2001). A liquid dispensing method and apparatus has a system with new individual dispensing heads connectible one each to a plurality of different liquid bottles, i.e. liquor, and a remote dispensing data receiver and computer that receives data from each head. Each head has a structure for being secured to a bottle, a liquid bore and an air vent, and one or more of the following features and functions: an

electronic dispensing timer, a stop pour annunciator, a magnetically latched dispensing control valve, an electronic bottle lock, a radio transmitter and antenna, an electronic fractional pour annunciator, programmable dispensing control, a micro-processor computer, a data storage, a data I/O structure, and structure and function for uniquely electrically identifying each head and liquid. The recover/computer has a data receiver and has structure and function to electronically program each head, the computer provides a record of all important dispensing data including head connection to bottles, head numbers, liquid identities, quantities of dispense cycle, quantity of liquid dispensed, inventory status and other desirable business data. The annuciator equipped head has an interior light that illuminates the entire transparent or transluscent head.

Another article by Credle Jr. (2002) explains postmix valve for a beverage dispenser, including a volumetric ratio control device incorporated therein to provide positive ratio control. The device includes a syrup piston and a soda piston linked together, syrup and soda chambers, and valve means for controlling the flow to and from the chambers. The soda pressure drives the pistons. The valve means preferably includes four solenoid valves for the water circuit and four one-way valves and a pressure regulator for the syrup circuit. The valve includes means for varying the total flow rate of the beverage being dispensed.

In the article by Yaxin (2002) a high speed MEMS flow sensor was proposed to enhance the reliability and accuracy of a liquid dispensing system. Benefiting from the feedback of sensor information, the system can self-adjust the open time of the solenoid valve to accurately dispense the desired reagent volume without pre-calibration. This paper focused on the design, fabrication and application of this flow sensor. Firstly, the design, fabrication and characteristics of the MEMS flow sensor based on the measurement of the pressure difference across a flow channel were presented. Secondly, the liquid dispensing system in which the flow sensor is integrated was introduced. A novel closed-loop control strategy was proposed to calculate valve open-time for each dispensing cycle. Finally, experiments results were presented with different dispensing volumes, coefficient of variance (CV) has been shown to be below 3% at 1μl and approach 4% at 100 nl. It indicated that integration of the MEMS flow sensor and using of a compound intelligent control strategy made the system immune to liquid viscosity, pressure fluctuation and some other disturbances.

Another article by Yao and Chen discusses a robotic liquid handling system. It was developed for dispensing a highly viscous reagent with nanoliter volumes. The robot in question was of immediate need in protein crystallization research and in the electronics packaging field. In this paper, the system structure was introduced which mainly consisted of three modules: motion control module, dispense control module and droplet volume measure module. Highly viscous reagent could be dispensed in nanoliters through controlling the

dispense control module and the motion control module correctly, and the volume of micro-drop could be measured based on a robotic vision technique. The factors that influence the successful delivery of nanoliter volumes of highly viscous reagent were discussed through analysis of the dispensing process. And the two critical values that the dispense height should be kept were derived. Finally, three kinds of reagent with different viscosity was used for dispensing experiments to verify the theoretical results. The accuracy of the system was shown to be below 7%, and the coefficient of variance (CV) has been shown to be below 10%.

According to an article published in the IEEE conference by Sawicki, titled Pump Dispensing Mechanism, liquid is typically contained in a container having a pump actuator head that is depressed to dispense the product onto the hand of the user. The containers can be in a wide variety of shapes, and there are different actuator heads and pumping means available, but the dispensers all operate on similar principles, with the actuator head being depressed, the product being drawn up a feed tube and dispensed through a spout or nozzle in the actuator head and onto the user's hand. The dispensers are generally simple and convenient to use but can cause problems when a consumer tries to operate the actuator head and dispense the contents with only one hand, with the other hand being unavailable to provide support to the dispenser, possibly because of stickiness, greasiness or other problems, or simply because the user simply desires to use only one hand for dispensing the composition. In particular, many

of the designs of dispenser containers are not sufficiently stable, especially when they have been emptied to a significant extent, to enable a consumer to operate the actuator head without using the other hand to support the dispenser to prevent it tipping or moving during operation of the actuator head. Therefore, there is a need in the art for a pump dispenser that allows effective one-handed operation of a fluid dispenser. The invention is a pump dispenser for one-hand operation. Current fluid pumps are designed to be pushed down with the palm of one hand while holding the other hand under the nozzle. Thus, one hand is used to dispense the fluid and the other hand receives the dispensed fluid. The present invention allows one handed operation of the pump dispenser because a user can push down with the back of the fingers and dispense the fluid into the same hand. One-handed operation of the present invention allows the user's other hand to be free for other tasks. The pump dispenser has an actuator head for the dispensing of a high viscosity fluid, such as soap or lotion. The actuator head is actuated through the use of a handle having at least two depression members. A user places his or her hand near the actuator head palm-side up, with the palm beneath the dispenser spout, and simultaneously places a finger on each handle depression member. The user uses his or her fingers to depress the handle to dispense the fluid. The structure of the actuator head and the handle conveniently allow for one-hand operation. As an example, the pump dispenser may be connected to a container to form a fluid dispenser.

In the article Liquid dispensing system with enhanced mixing by Belongia and Saunders (2003) a dispensing apparatus and system for dispensing suspensions or emulsions is discussed. The system ensures uniformity of distribution of the dispersed phase within the continuous-phase liquid by moving the fluid through the dispense cartridge, such as with a continuous or pulsating flow. In one embodiment, peristaltic pumps are positioned upstream and downstream of the dispense cartridge, in fluid communication with and forming a single loop with a fluid source. Circulation between the fluid source and the dispense cartridge is maintained. In a second embodiment, a pump circulates fluid into and out of the dispense cartridge and is also in fluid communication with a fluid source such as with a pinch valve to allow proper filling of the dispense cartridge from the fluid source. In a third embodiment, a reversing pump is placed between the dispense cartridge and fluid source to continually or continuously pump fluid into and out of the dispense cartridge.

Chapter 3

DESIGN PROCEDURE

This chapter gives a detailed discussion of the step-by-step procedures used in developing the design topic. This section discusses hardware development (conceptual, block and schematic diagram), and the materials and components used.

Design Procedures

Researchers must follow these procedures in order to create the design. These steps are recorded as follows:

1. **Problem Definition**

    The group identified the main problem that needed to be solved. After which the group tried to construct a vague concept of the system that was designed. After constructing the concept, the group identified the functions, objective and scope and delimitations of the design. Taking these steps made the design process easier to create.

2. **Data and Requirements Gathering**

    The group gathered data from different resources related to the design. They consulted people who have broad knowledge of the topic and also turned to books and journals and other means of references.

3. Design Planning

The data gathered served as the basis for the beginning of the designing process. The group then determined all the requirements that were needed for the process of designing.

4. Hardware Development

4.1 Block Diagram

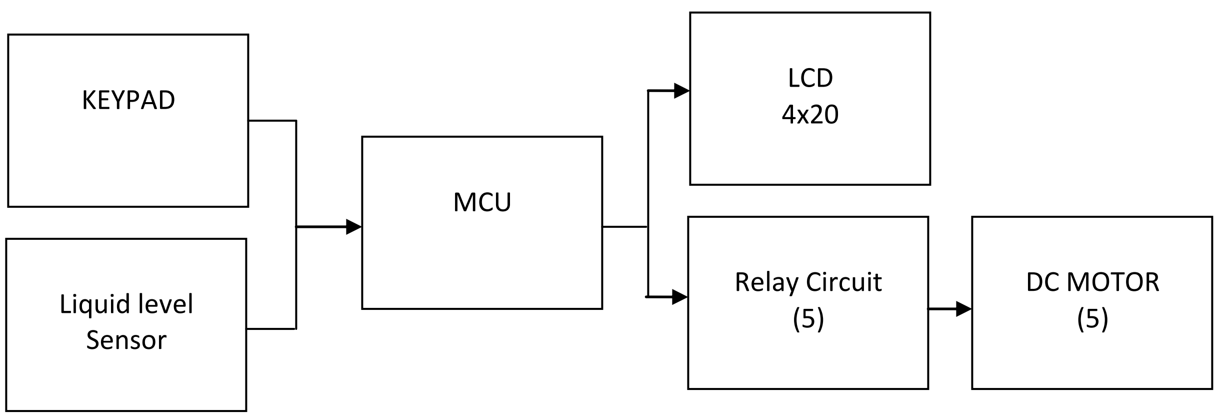


Figure 3.1 Block Diagram

Fig 3.1 shows the interconnections between the components of the system. The design is composed of seven components, namely the keypad and the liquid level sensors (which serve as the input), microprocessor unit (processor), and the LCD and the relay circuits which control the DC motor (output). These are powered by a full wave center tap power supply. The keypad allows users to enter the drink they wish to dispense. Then the liquid level

sensor will determine if the amount of liquid inside the containers is still sufficient enough to make the drink and will send the result to the microprocessor. If the sensor detects that the amount of liquid is below 5% of the container it will no longer allow the system to dispense the drink and the user will be notified by displaying the message on the LCD. Otherwise, the microprocessor is going to dispense the drink if the amount of liquid is sufficient enough to make the drink. The relay will be activated and the DC motor will pump the liquid to be dispensed. The microprocessor processes the input and output.

The components of the design are the following:

a. PIC16F877A controls all the processes running inside the design. It has 40 pins, 33 programmable IO and an 8k size programmable memory.
b. The C9018 transistor is connected to the liquid probes which serve as the liquid level sensor and one of the inputs of the microprocessor. It senses if the liquid level is at a low level so that the system will no longer dispense. There are LEDs that are connected to let the user be aware that the water level is running low or close to empty.
c. The keypad is where the user can access, operate the machine. It allows the user to interact with the system. The keypad is in a matrix form inside.
d. LM044L is a 20 character x 4 line LCD, which displays data to the users.
e. ULN2003 is the relay driver responsible for triggering the relay. It is controlled by the microprocessor. The microprocessor can only output up

to 5V of power so the relay is connected to give 12V of power to switch the motor on or off.

f. The DC motor is used to pump the liquid from the container, to be dispensed. There are five motors placed in the system for each of the five containers.

4.2 Schematic Diagram

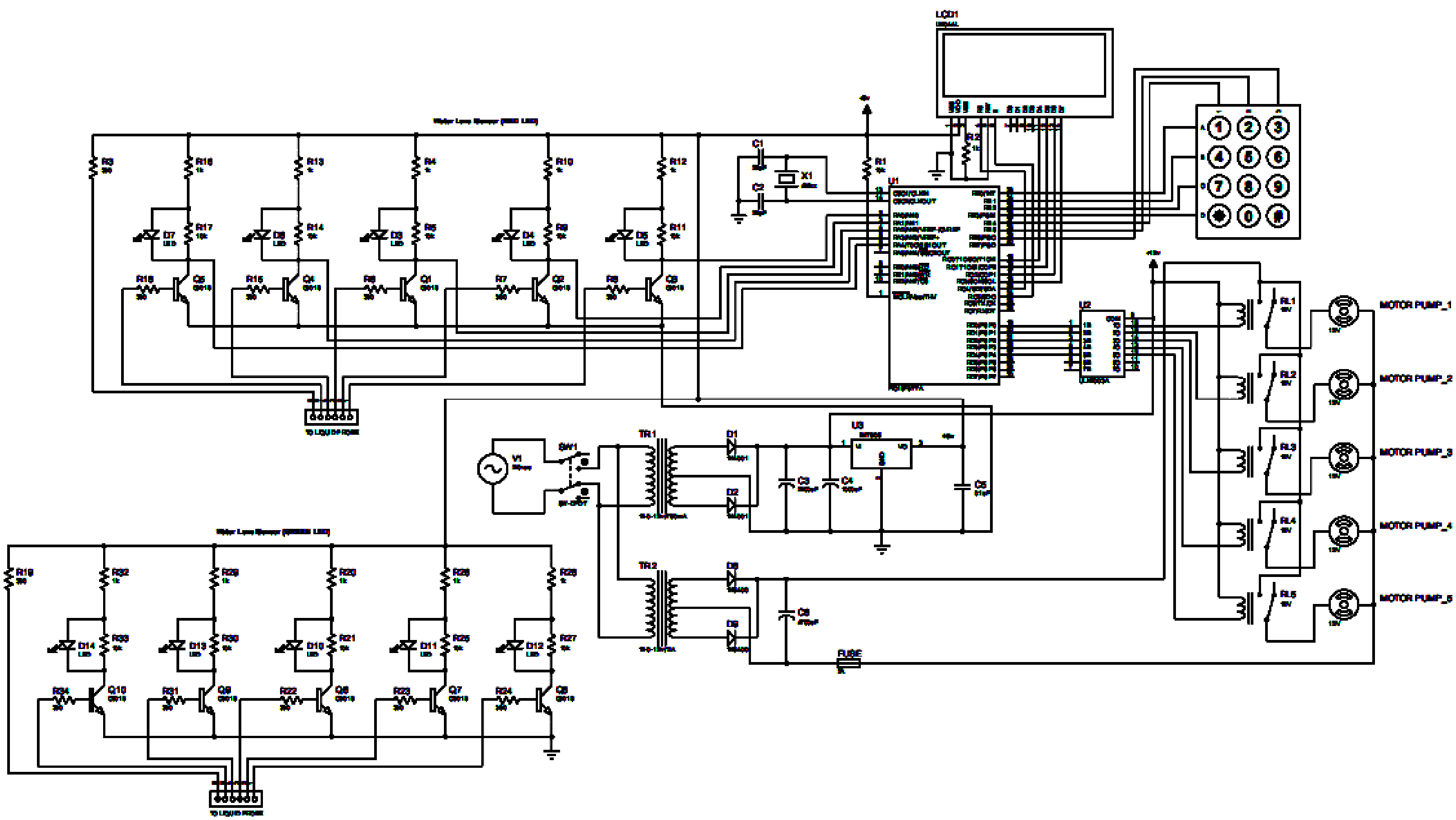


Figure 3.2 Schematic Diagram of Automated Liquor Dispenser using PIC16F877 with Liquid Level Sensor

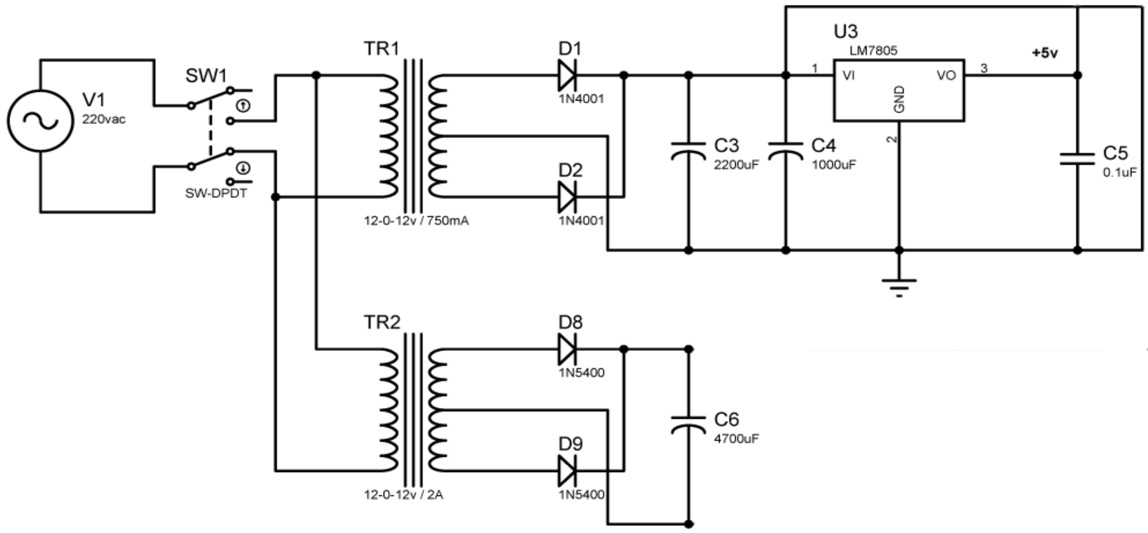


Figure 3.3 Power supply

The circuit consists of two diodes, each one connected at opposite ends of the transformer windings. The diodes alternately conduct. For a complete cycle of the AC voltage waveform, one diode would conduct at the positive portion of the cycle and the other at the negative portion. Both half cycles, however, would be positive with respect to the center tap, since each cycle is positive with respect to the center tap of the transformer. The transformer if intended for electronic application is always assumed to have a 92% efficiency.

The power supply supplies 12V to the relay and the ULN2003 relay driver. An IC voltage regulator unit, LM7805, which takes a DC voltage and provides a lower DC voltage, was connected to the power supply to reduce the 12V output to 5V. The 5V was used to supply for the microcontroller and the collector side of the transistor as well.

Placing capacitors in LM7805 is optional, although it improves transient response. A 1µF output capacitor was placed. Based on the datasheet of LM7805 output capacitors are commonly used to provide improved output impedance and rejection of transients.

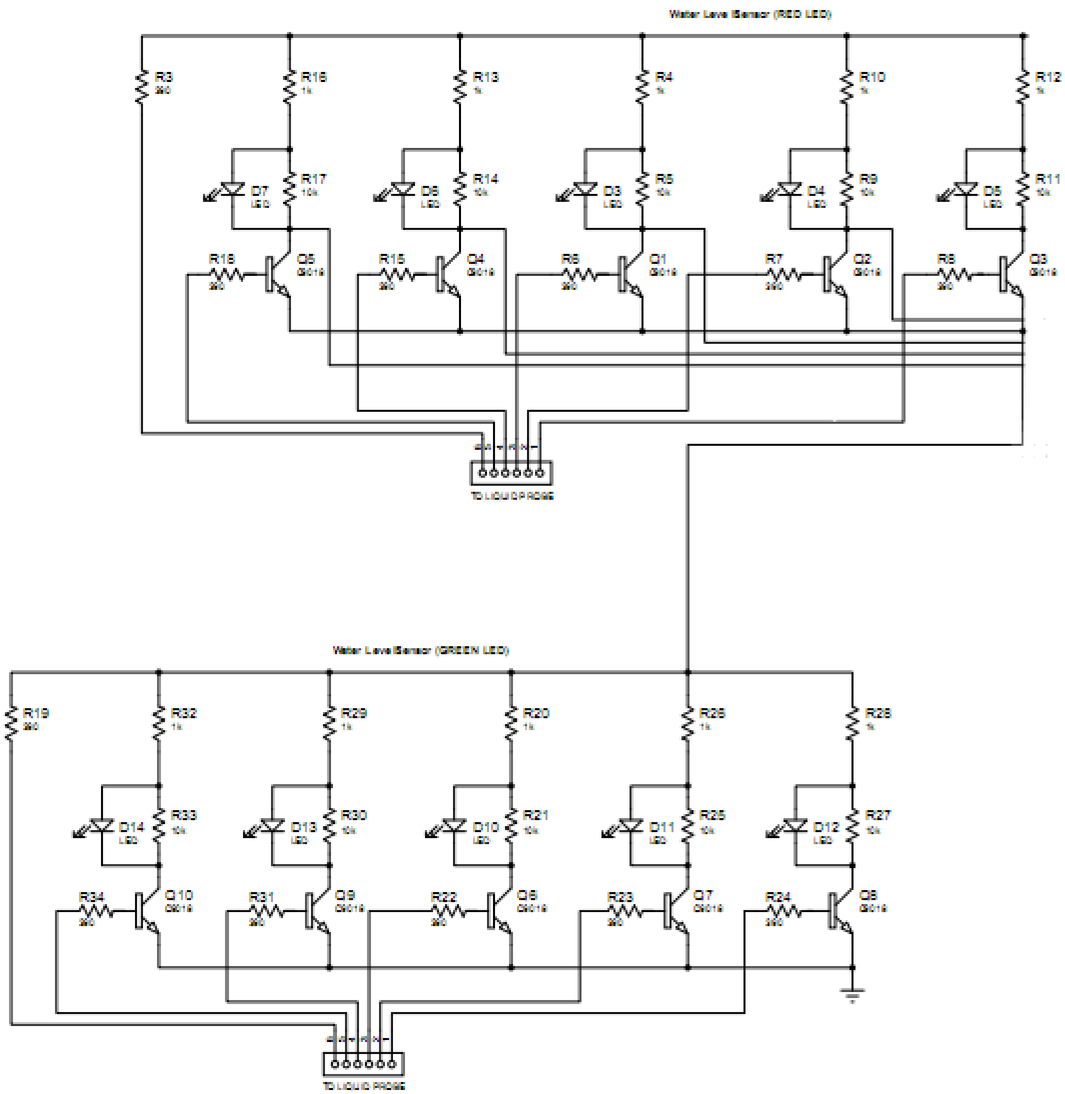


Figure 3.4 Liquid Level Sensors

The liquid level sensor detects if there is still liquid inside the container. If there is liquid the base and collector side of the transistor will short and current will pass causing the LED to light up and send a positive signal to the microprocessor. If the container is empty, the emitter will send a ground signal to the collector and the LED will not light up. Resistors are placed on the base side of the transistor to limit the current going to the transistor. A 390Ω resistor RB is connected to the base of NPN transistor Q1-Q10 to avoid damage. The maximum base current that the transistor could handle is 12mA. R6-8,R15-18, R22-24, R31, R34 can be computed as shown in the equation

$$R_B = \frac{5V - 0.7V}{12mA} = 358.33\Omega$$

Equation 3.1 Value of R_B

Since there are no available resistors of the computed value the group decided to use a 390 Ω resistor.

On the collector side, there are two resistors that are placed. The total value of the resistor is 11K Ω . The 1K Ω resistor is used just to limit the 5V that will pass through the transistor. The 10 K Ω resistor is placed in parallel with the LED diode. The diode is off when the liquid level is low or empty otherwise it is turned on. The group placed a 10K Ω resistor so that the current can pass through it when the liquid level is low. The 10 K Ω resistor serves as a pull-up resistor.

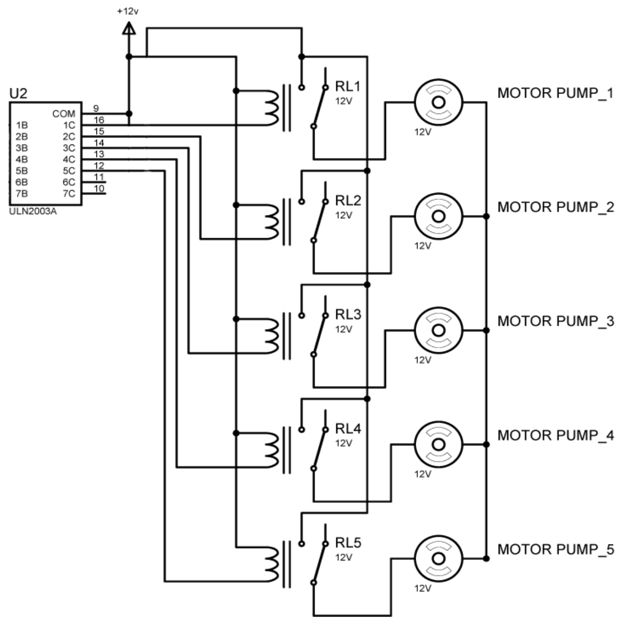


Figure 3.5 Relay Circuit and DC Motor

The relay circuit is composed of the relay driver (ULN2003A) and relays. DC motor pump is already included in figure 3.5. Since the microcontroller can only output 5V, the group included relays so that it can supply for the DC motor which requires a 12V input. The relay driver is controlled by the microcontroller. It is an IC that is a seven Darlington array that acts as a driver for the relays. Pins 19-22 and 27 of the microcontroller are connected to pins 1-5 of the relay driver. The output pins 12-16 are connected to the relays. If the relay is switched on the pump will turn on, too.

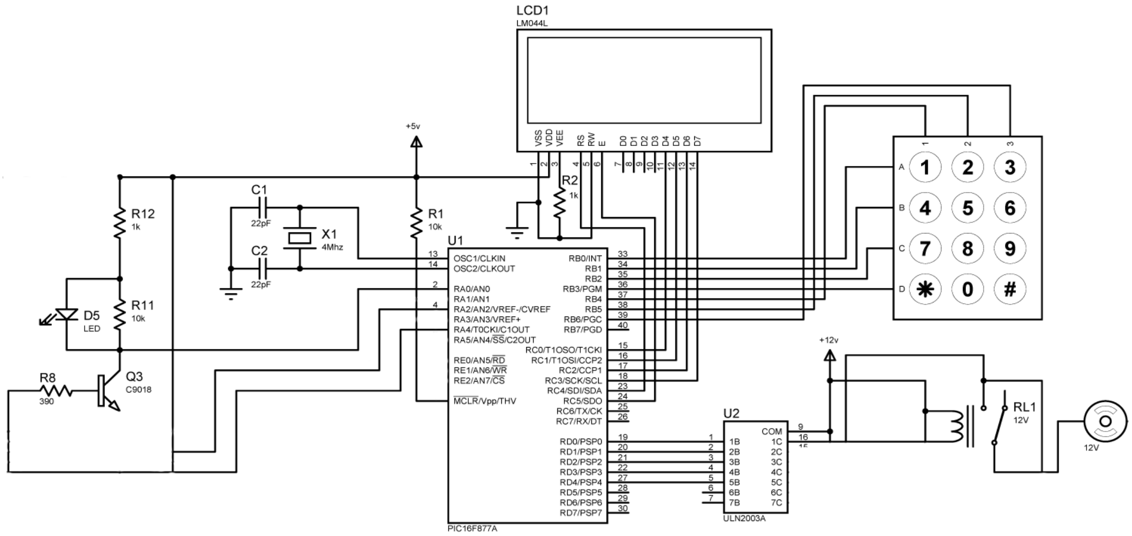


Figure 3.6 Microprocessor

Figure 3.6 shows the microcontroller, a 4 MHz crystal with two capacitors C_1 and C_2. Based on the requirements found on the PIC16F877A data sheet, 4 MHz frequency and capacitance of C_1 and C_2, with a value of 22pF, were used.

According to the PIC16F87X sheet, the maximum weak pull-up current is 500μA. Having 5V, the value of the pull-up resistor can be computed using Ohm's law.

$$R1 = \frac{V}{I} = \frac{5V}{500\mu A} = 10K$$

Equation 3.2 Value of R_1

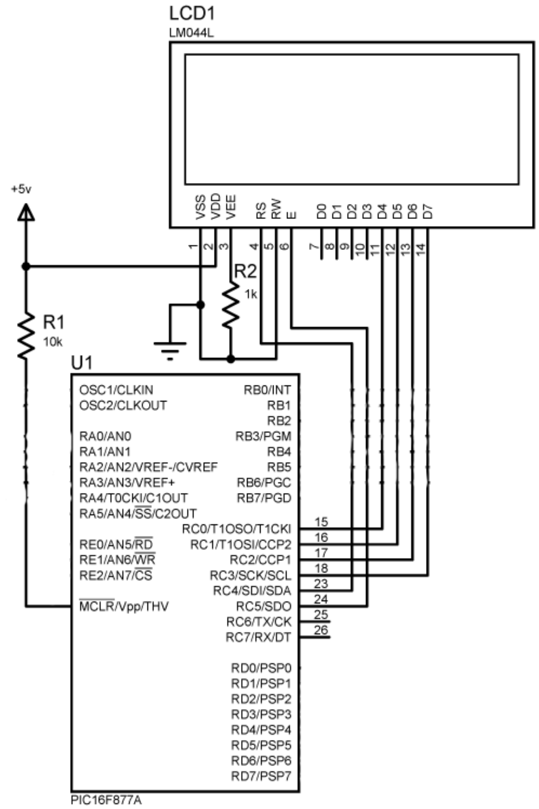


Figure 3.7 LCD

LM044L is a 20 character x 4 line LCD, and displays data to the users. The first pin of the LED is connected to ground. Pin 2 is the supply pin, connected to a 5V supply. Pin 3, Vee is for contrast connected to ground. Pins 4-6 are control pins. Pin 4 is the register select pin that is connected to pin 23 of the microcontroller. Pin 5 is the read/write pin. Since the system only does the write operation pin 5 connected to ground. Pin 6 is the enable. Pins 12-14 are the data pins where in the data that are to be displayed. It is connected to pins 15-18 of the microcontroller. The 1K resistor is a requirement of the LCD; this will provide an easily variable voltage between Ground and Vcc, which will be used to specify the contrast (or "darkness") of the characters on the LCD screen.

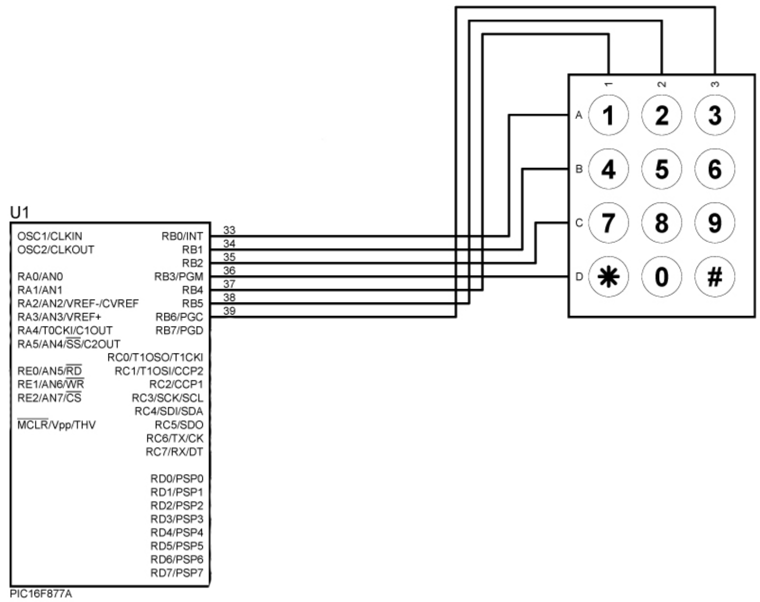


Figure 3.8 Keypad

The keypad is programmed and is in matrix form inside. It is connected to the microcontroller. Pins 33-39 of the microcontroller are connected to the keypad.

5. Software Development

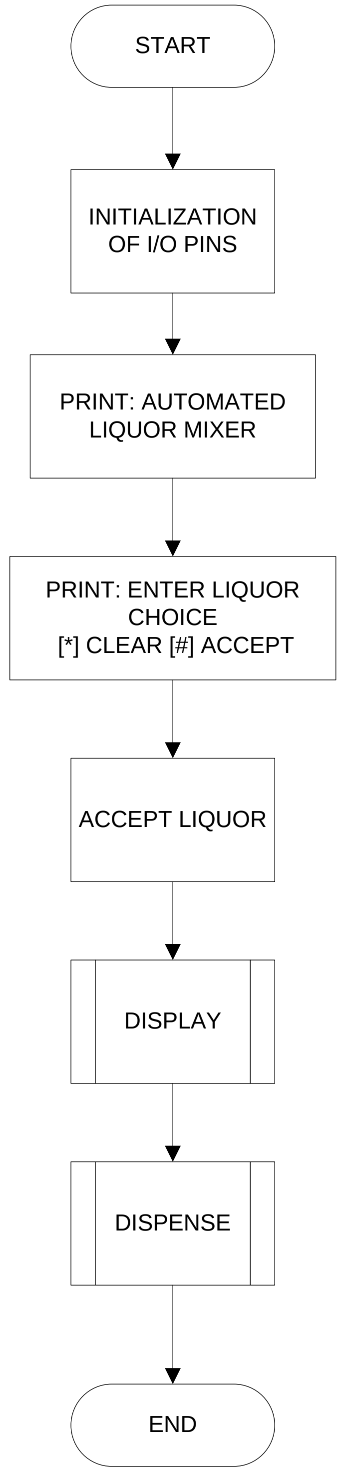


Figure 3.9 System Flowchart of Automated Liquid Dispenser using PIC16F877 with Liquid Level Sensor

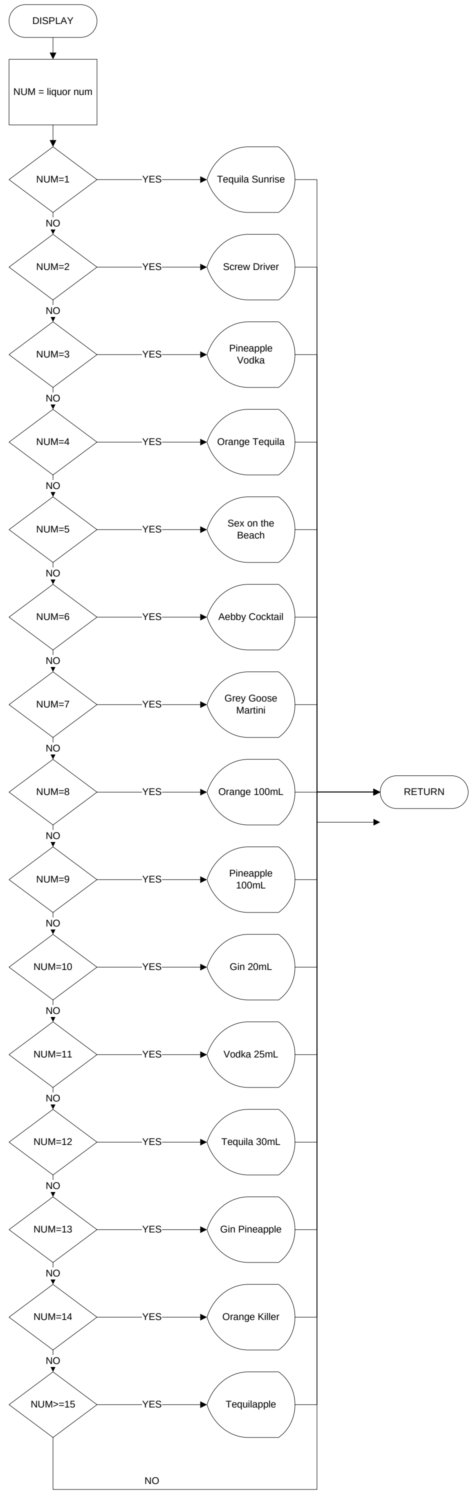


Figure 3.10 Display Sub-function Flowchart

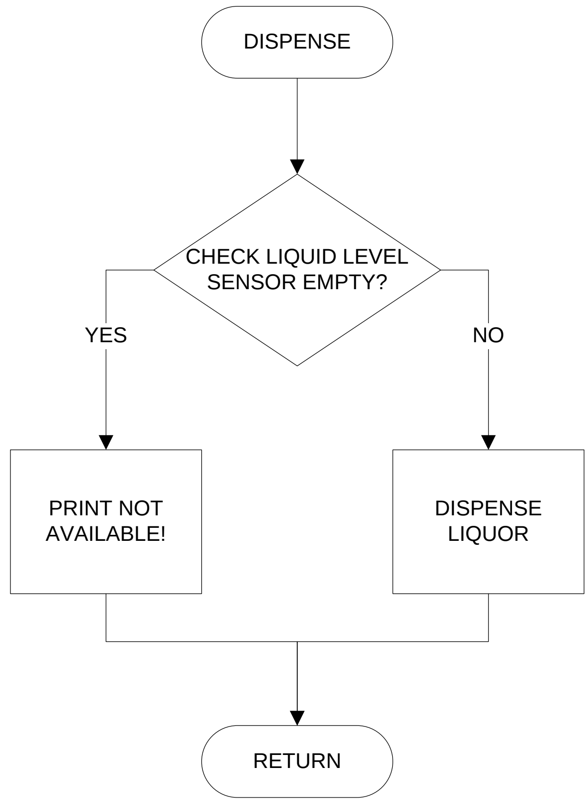


Figure 3.11 Dispense Sub-Program

Figure 3.9 shows the flowchart of the main program of the microcontroller. PIC Basic was used to program the PIC16F877A IC. Basically, the logic behind the program is that each relay was set to a specific time delay (for every 10 mL there is a 600msec time delay.) which tells the amount of liquid to be dispensed. It also includes the LCD which tells it what to display, and it interprets the input from the keypad and the liquid level sensor.

6. Prototype Development

Prototype development was made after accomplishing the hardware and software components of the design. The components were first tested out on a breadboard then etched, mounted and soldered on a PCB. The designers paid close attention to the polarities and placement of the components. PCB circuits were placed above the storage in case of possible leaks coming from the containers placed above. The figures below show the circuit layout of the Relay PCB Layout (Figure 3.12), Power Supply Layout (Figure 3.13), and Microprocessor PCB Layout (Figure 3.14).

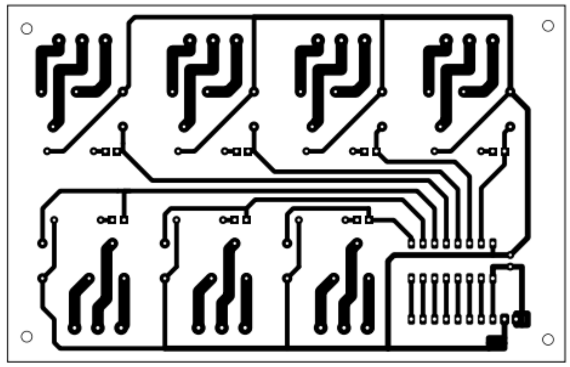


Figure 3.12 Relay PCB Layout

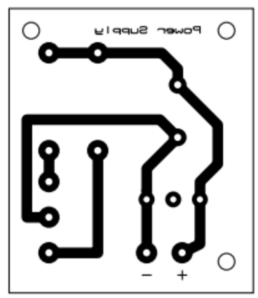


Figure 3.13 Power Supply PCB Layout

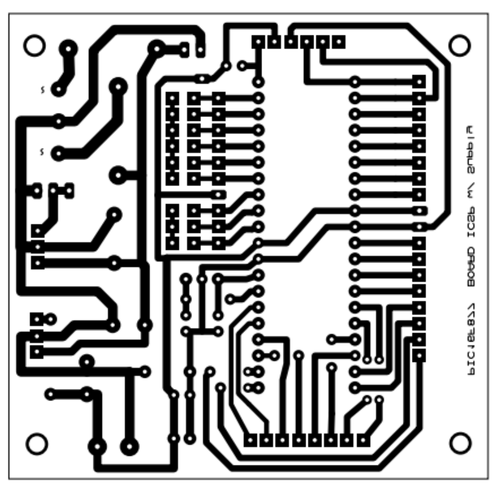


Figure 3.14 Microprocessor PCB Layout

#	Materials	Function
1	ALUMINUM CASING	Used as the housing or outer covering of the circuit. It protects the circuit from external forces.
2	DC MOTOR PUMP	Used to deliver forth the liquid from the container by means of suction. It is connected to the relay.
3	LCD	Used as an interface between the user and the device. It has 20 characters with 4 lines and powered by a 5V supply
4	KEYPAD	One of the inputs of the device. It allows the user the combination corresponding to the liquor of his/her choice.
5	LIQUOR CONTAINER	Used to hold the basic ingredients for the liquor mixes.
6	LIQUID LEVEL SENSOR	The other input that is used to detect the liquid level.
7	12V RELAY	Receives the signal from the relay driver and passes it to the DC motor so that the liquid can be suctioned and passed on to be dispensed.
8	MCU PIC16F877	Controls all the processes running inside the device. It has 40 pins, 33 programmable IO and an 8k size programmable memory.
9	LED	Part of the sensor and serves as an indicator of the liquid level for the user.
10	ULN2003A	The relay driver responsible for triggering the relay. It is controlled by the microprocessor, which can only output up to 5V of power so the relay is connected to give a 12V power to switch the motor on or off.
11	POWER SUPPLY	Supplies 12V to the relay and the ULN2003, relay driver. An IC voltage regulator unit, LM7805.
12	PCB	Used to mechanically support and electrically connect electronic components using conductive pathways.

Table 3.1 Components Used for the Prototype

QTY	MATERIALS	UNIT	PRICE
1	ALUMINUM CASING	5,500	5,500
5	DC MOTOR PUMP	700	3,500
1	LCD	1,200	1,200
1	KEYPAD	160	160
5	LIQUOR CONTAINER	500	2,500
10	LIQUID LEVEL SENSOR	50	500
5	12V RELAY	55	275
1	MCU PIC16F877	300	300
1	40 PINS IC SOCKET	15	15
3	8PIN HEADER CONNECTOR	34	102
1	PUSH BUTTON	20	20
1	PHOTO SENSOR	200	200
2	MICRO-SWITCH	25	50
2	LED	5	10
10	RESISTOR	0.25	2.5
1	ULN2003A	25	25
1	2PIN TERMINAL BLOCK	14	14
1	POWER SUPPLY	700	700
1	AC PLUG	15	15
1	AC POWER CORD	50	50
-	PCB	350	350
-	MISCELLANEOUS	3,000	3,000
TOTAL			**18,506.50**

Table 3.2 Continuation of List of Components

7. Hardware and Software Implementation

In integrating the hardware and software, both components must be properly tested and ensured that both are functioning. The design must be compatible with each other. The designers ensured that all the objectives are met by the design.

8. Testing and Troubleshooting

The group tested and tried to locate and fix the possible sources of trouble for the design. During the testing part, the group discovered that the amount of the dispensed liquid was not consistent. The group fixed the problem by reprogramming the PIC and changing the time delay for each input. By doing so, the group was able to fix the problem.

9. Final Testing

Final testing of the design was made to ensure that the design is working properly, the objectives are met and the scope is covered fully.

Chapter 4

TESTING, PRESENTATION, AND INTERPRETATION OF DATA

This chapter gives a detail on how the system was tested in accordance with the objectives stated in Chapter 1. The testing was done after development and construction of the prototype to verify if the objectives were achieved. In addition, the following tests determined the functionality and reliability of the created device.

LIQUID LEVEL SENSOR TEST

A liquid level sensor was placed inside the container together with the pumps. Its operation was based on the shorting of two pins; that is, if the liquid was in contact with the two pins, they were shorted otherwise, they were open. If one of the water level sensors was open, the system informed the user via an LED indicator. To signal the user which one of the containers had insufficient liquor required for an operation, an array of LED labeled from one (1) to five (5) were placed.

The following steps were done to execute the test:

1. Fill at least 75% of all the containers with their corresponding liquid ingredient.
2. Start by emptying a single container to emptying several containers by testing all the combinations of liquor mixes.

When one of the containers was emptied or its contents were set below the minimum liquid level, its corresponding LED indicator will turn on. When two or more containers were emptied or their contents were set below the minimum liquid level, their corresponding LED indicators would turn on.

DISPENSING ACCURACY TEST

Dispensing the exact amount of liquid for each kind of liquor mix to be processed by the machine was very important since it significantly affected the taste of the end product.

The following steps were done to execute the test:

1. Each container must have an adequate amount of liquid ingredients for the design to dispense the liquor mix accurately.
2. Test the first liquor mix combination was test in 5 trials.
3. Step 2 was done until all combinations were tested.

Liquor Mix	1	2	3	4	5
Tequila Sunrise		1.87%			1.65%
Screw Driver	5.48%			1.24%	
Pineapple Vodka		3.45%		1.32%	
Orange Tequila	4.89%				1.78%
Sex On The Beach	3.50%	2.34%		1.28%	
Aebby Cocktail			0.44%		
Grey Goose Martini	1.80%	2.45%		2.10%	
Gin Pineapple	2.50%		0.44%		
Orange Killer	1.24%		0.44%	2.25%	
Tequila Apple		4.45%			1.34%

Table 4.1 Average Errors of Dispensed Ingredients in Liquor Mixes

Table 4.1 shows the average error in percentage out of 5 trials of each dispensed liquid for a specific kind of liquor mix produced. From the results, it was observed that the densest of the six main liquid ingredient was Orange Juice (2) exhibiting a list of errors ranging from 1.24% to 5.48%. Second was the Pineapple Juice (3). The least dense was the Gin which exhibited errors not more than 1.00%. It can also be concluded that different juice types are dense in nature as shown through the dispensing behavior of pineapple juice and orange juice.

Chapter 5

CONCLUSION AND RECOMMENDATION

This chapter states the overall conclusion of the design which addresses if the objectives were actually met.

Conclusion

The design is capable of dispensing drinks that are programmed in its microcontroller. The design helps reduce manual labor behind the bar counter by dispensing drinks.

The researchers were able to optimize the utilization of components such as the microcontroller, relays, and DC motors.

In relation to the objectives and testing done, the researchers were able to conclude the following:

a. The design is a machine that can help a bartender in making an alcoholic drink. It can dispense drinks chosen by the user.
b. The design is capable of dispensing close to the exact amount of ingredients for all drinks programmed in it. The density of the liquid is taken to in account considering that the highest percentage error, obtained from the results of the design testing, was exhibited by the liquid with the highest density (Orange juice). The lowest percentage error, not more than 1.00%, was exhibited by the liquid with the lowest density (gin).

c. The design is able to produce numerous drinks, provided that the container is still not empty, without altering the quality.
d. The design prevents foreign matter from mixing with the drinks due to minimal human intervention. The liquid level sensor prevents the operator from opening and checking if the amount of liquid is still sufficient enough to dispense drinks.

Recommendation

Further enhancement of the design could be done for the improvement of its functionality and to optimize resources. The design can be enhanced by making it into a fully utilized vending machine that already accepts currency, thereby fully eliminating the need for man power. Such a machine could dispense glasses on its own, even adding ice or additional cocktail ingredients. Also, an additional base container can be added to increase the number of liquor mixes.

The design could be improved by instead of using a DC motor for dispensing, a flow meter could be used. So that the liquid dispensed would be more accurate.

It could also be further developed by incorporating a cooling system. Also, a stirrer can be added so that the drinks produced can be mixed evenly. An additional microprocessor can make the additional stirrer possible because the researchers have already optimized the full use of the microprocessor used.

The design could also be improved by allowing the user to choose a drink that is not programmed in the system. For now, if additional mixes are asked, the microprocessor has to be detached and reprogrammed.

BIBLIOGRAPHY

Belongia, B. and Saunders, R. (2003), Liquid Dispensing System with Enhanced Mixing, *IEEE Conference* pp 102-109.

Desa, D. (2004) *Applied Technology and Instrumentation for Process Control*, New York, New York:

Garvie, C. (2002), Liquid tot Dispenser, *IEEE conference,* pp 501-520.

Goulet, D. and Bartoletti, J. (1999), Beverage Dispenser, *IEEE International Workshop* , pp 81-93.

Jinhuang, H. (2003), The Intelligent Water Dispenser System based on Embedded Systems, *IEEE Conference,* pp 680-691.

Johnson, C.D. (1997). *Process Control Instrumentation Technology*, 5th edition. New Jersey:

Nashelsk, R. (2004) *Electronic Devices and Circuit Theory 7th Edition*, New Jersey:

Peckels, A. (2001), Electronic Dispensing Head, *IEEE International Workshop*, pp 134-143

Yaxin, L. (2002), MEMS Flow Sensor and Its Application in Adaptive Liquid Dispensing, *IEEE international workshop,* pp 120-125.

APPENDICES

APPENDIX A

Operation's Manual

1. **User's Manual**

   a. Plug into a 220Vac source

   b. Turn on the device.

   c. Place a glass (at least 150ml) below on the tray.

   d. Input the number corresponding to the desired liquor (choose from

   the

   list of different liquor mixes)

   e. If the number entered is wrong, press clear * sign.

   f. Then press # sign to dispense the liquor.

2. List of liquor mixes

a. Tequila Sunrise
   i. 50ml tequila
   ii. 40ml orange
   iii. 10ml pineapple
b. Screw driver
c. vodka
   i. 50ml orange
d. Pineapple vodka
   i. 70ml pineapple
   ii. 30ml vodka
e. Orange tequila
   i. 40ml tequila
   ii. 60ml orange
f. Sex on the beach
   i. 20ml vodka
   ii. 40ml orange
   iii. 40ml pineapple
g. Aebby cocktail
   i. 55ml gin
   ii. 45ml orange

h. Grey goose martini
    i. 50ml vodka
    ii. 40ml orange
    iii. 10ml pineapple
i. 100ml orange
    i. 100ml pineapple
    ii. 20ml gin
    iii. 25ml vodka
    iv. 30ml tequila
j. Gin pineapple
    i. 40ml gin
    ii. 60ml pineapple
k. Orange killer
    i. 15ml vodka
    ii. 15ml gin
    iii. 70ml orange
l. Tequilapple
    i. 50ml tequila
    ii. 50ml pineapple

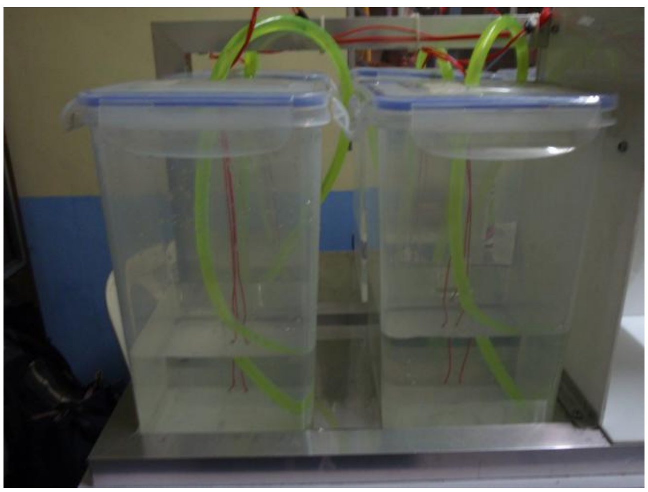
3. Troubleshooting

Consult the table if the user is experiencing technical difficulties.

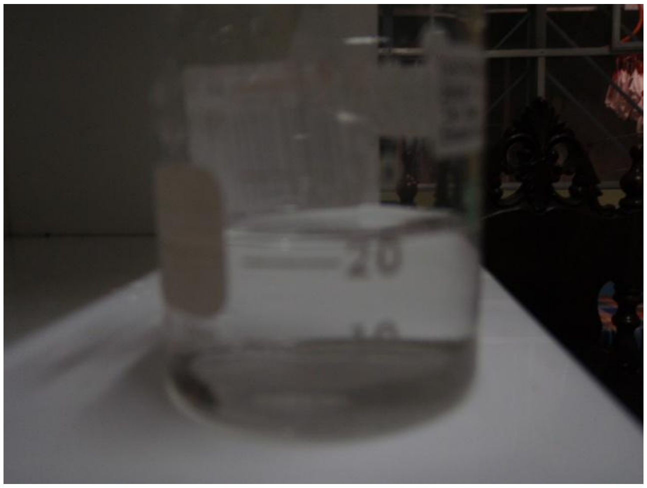
Problem	Suggestion
Liquid not dispensing properly	Check the sensor that corresponds to each container. If the LED is not turned on, the container is empty. Refill the device and run it 2-3 times so that it will dispense the right amount.
	If the device has just started dispensing and the liquid being dispensed is not the right amount, run the device 2-3 times by dispensing liquid from each of the 5 containers. This way the hose will fill up and start dispensing the right amount.
Not dispensing at all	Check if device is plugged into a 220Vac.
	Check the sensors if containers still have liquid in it.
	Check if the switch is turned on.

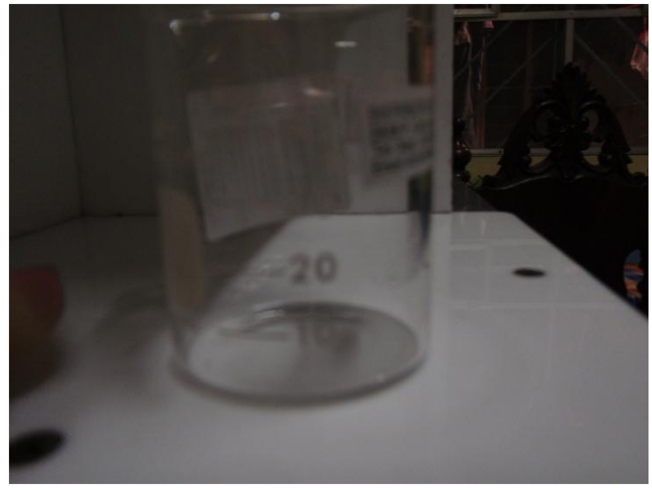
APPENDIX B

Pictures of the Prototype



Testing Pictures





APPENDIX C

Program Listing

```
'TITLE: Coctail Mixer
'DATE : 04/05/11

'Components:
' - LCD
' - Keypad
' - Relay
' - Motor pump

Device 16F877A          ' MCU TYPE
Xtal = 4                ' 4MHZ OSCILLATOR
LCD_DTPin = PORTC.0          ' Used for 4 data line interface.
LCD_RSPin = PORTC.4          ' Assigns the Port and Pins that the LCD's RS
                               line will attach to.
LCD_ENPin = PORTC.5          ' Assigns the Port and Pin that the LCD's EN
                               line will attach to.
LCD_Interface = 4            ' Inform the compiler 4-line interface is
                               required by the LCD.

LCD_Lines = 4                ' Inform the compiler how many lines the LCD
has.
LCD_Type = 0                 ' Inform the compiler that the type of LCD is
                               alphaNUMERIC.

All_Digital = On
PortB_Pullups = On      ' SET INPUT NORMAL LOGIC HIGH

''''''''''''''''''''''''''''''''''''''''''''''''''
'       - INITIALIZATION OF I/O PINS-
''''''''''''''''''''''''''''''''''''''''''''''''''
TRISA = %111111         ' LIQUID LEVEL SENSOR
TRISB = %01110000       ' KEYPAD
TRISC = %00000000       ' LCD
TRISD = %00000000       ' RELAY CIRCUIT / MOTOR PUMP

''''''''''''''''''''''''''''''''''''''''''''''''''
'          - VARIABLE DECLARATION-
''''''''''''''''''''''''''''''''''''''''''''''''''
Dim key  As Byte        ' KEPAD VALUE
Dim vol  As Byte        ' NUMBER OF VOLUME
Dim var1 As Word        ' VARIABLE
```

```
   Dim liquor As Byte     ' Liquor type
   Dim T1 As 2000
   Dim T2 As 200

   Symbol PUMP_1 = PORTD.0
   Symbol PUMP_2 = PORTD.1
   Symbol PUMP_3 = PORTD.2
   Symbol PUMP_4 = PORTD.3
   Symbol PUMP_5 = PORTD.4

   PORTB=0                ' clear ports
   PORTC=0
   PORTD=0
   DelayMS 500            ' 0.5sec delay

BEGIN:
   Cls                                                    ' Clear the LCD
(also creates a 30ms delay)
   Print $FE, $0C                         ' CURSOR OFF
   Print At 1, 1, "********************"   ' Place the text on line 1, position 1
   Print At 2, 1, "*     AUTOMATED    *"   ' Place the text on line 2, position 1
   Print At 3, 1, "*   LIQUOR MIXER  *"  ' Place the text on line 3, position 1
   Print At 4, 1, "********************"   ' Place the text on line 4, position 1
   DelayMS 3000

SELECTION:
   Cls
   Print $FE, $0C
   liquor = 0
   Print At 1, 1, "    ENTER LIQUOR   "
   Print At 2, 1, "   CHOICE: ",Dec liquor
   Print At 4, 1, "[*]CLEAR   [#]ACCEPT"
   DelayMS 500

  loop:
   GoSub KEYPAD
   If key = 11 Then key = 0
   If key = 10 Then SELECTION                  ' Clear
   If key = 12 And liquor > 0 Then DISPENSE   ' Accept
   If key > 9 Then loop
   liquor = (liquor * 10) + key
   If liquor > 15 Then liquor = 15    ' limit
   Print At 2, 1, "   CHOICE: ",Dec liquor
```

```
    Select liquor
    Case 1
      Print At 3, 1, " - Tequila Sunrise -"
    Case 2
      Print At 3, 1, "  - Screw driver -  "
    Case 3
      Print At 3, 1, " - Pineapple Vodka -"
    Case 4
      Print At 3, 1, " - Orange Tequila - "
    Case 5
      Print At 3, 1, "- Sex on the beach -"
    Case 6
      Print At 3, 1, " - Aebby Cocktail - "
    Case 7
      Print At 3, 1, "-Grey Goose Martini-"
    Case 8
      Print At 3, 1, "  - Orange 100mL -  "
    Case 9
      Print At 3, 1, " - Pineapple 100mL -"
    Case 10
      Print At 3, 1, "    - Gin 20mL -    "
    Case 11
      Print At 3, 1, "   - Vodka 25mL -   "
    Case 12
      Print At 3, 1, "  - Tequila 30mL -  "
    Case 13
      Print At 3, 1, "  - Gin Pinapple -  "
    Case 14
      Print At 3, 1, "  - Orange Killer - "
    Case 15
      Print At 3, 1, "   - Tequilapple -  "
    EndSelect

    GoTo loop

DISPENSE:
    PORTC = 0
    Select liquor
    Case 1
      GoSub SENSOR_4
      GoSub SENSOR_1
      GoSub SENSOR_2
      PUMP_4 = 1    ' tequila 50mL
      DelayMS 3000
```

```
  PUMP_4 = 0
  DelayMS T1

  PUMP_1 = 1    ' orange 40mL
  DelayMS 2450 '2178
  PUMP_1 = 0
  DelayMS T1

  PUMP_2 = 1    ' pineapple 10mL
  DelayMS 580
  PUMP_2 = 0
Case 2
  GoSub SENSOR_1
  GoSub SENSOR_3
  PUMP_3 = 1    ' VODKA 50mL
  DelayMS 3600
  PUMP_3 = 0
  DelayMS T1

  PUMP_1 = 1    ' ORANGE 50mL
  DelayMS 3000
  PUMP_1 = 0
Case 3
  GoSub SENSOR_3
  GoSub SENSOR_2
  PUMP_2 = 1    ' pineapple 70mL
  DelayMS 4300 '4900'5450
  PUMP_2 = 0
  DelayMS T1

  PUMP_3 = 1    ' VODKA 30mL
  DelayMS 2000
  PUMP_3 = 0
Case 4
  GoSub SENSOR_1
  GoSub SENSOR_4
  PUMP_4 = 1    ' TEQUILA 40mL
  DelayMS 2670  '2000
  PUMP_4 = 0
  DelayMS T1
  PUMP_1 = 1    ' ORANGE 60mL
  DelayMS 3900 '4000
  PUMP_1 = 0
Case 5
```

```
  GoSub SENSOR_1
  GoSub SENSOR_2
  GoSub SENSOR_3
  PUMP_3 = 1    ' VODKA 20mL
  DelayMS 1800
  PUMP_3 = 0
  DelayMS T1

  PUMP_1 = 1    ' ORANGE 40mL
  DelayMS 2450
  PUMP_1 = 0
  DelayMS T1

  PUMP_2 = 1    ' pineapple 40mL
  DelayMS 2400
  PUMP_2 = 0
Case 6
  GoSub SENSOR_1
  GoSub SENSOR_5
  PUMP_5 = 1    ' GIN 55mL
  DelayMS 3050
  PUMP_5 = 0
  DelayMS T2

  PUMP_1 = 1    ' ORANGE 45mL
  DelayMS 2450
  PUMP_1 = 0
Case 7
  GoSub SENSOR_1
  GoSub SENSOR_2
  GoSub SENSOR_3
  PUMP_3 = 1    ' VODKA 50mL
  DelayMS 3000
  PUMP_3 = 0
  DelayMS T1

  PUMP_1 = 1    ' ORANGE 40mL
  DelayMS 2200
  PUMP_1 = 0
  DelayMS T1

  PUMP_2 = 1    ' PINEAPPLE 10mL
  DelayMS 570
  PUMP_2 = 0
```

```
Case 8
  GoSub SENSOR_1
  PUMP_1 = 1    ' ORANGE 100mL
  DelayMS 6405 '5445
Case 9
  GoSub SENSOR_2
  PUMP_2 = 1    ' PINEAPPLE 100mL
  DelayMS 5791 '5731 '5445
Case 10
  GoSub SENSOR_5
  PUMP_5 = 1    ' GIN 20mL
  DelayMS 1250 '1150 '280
Case 11
  GoSub SENSOR_3
  PUMP_3 = 1    ' VODKA 25mL
  DelayMS 1800  '1500 '600
Case 12
  GoSub SENSOR_4
  PUMP_4 = 1    ' TEQUILA 30mL
  DelayMS 1900  '2000 '840
Case 13
  GoSub SENSOR_5
  GoSub SENSOR_2
  PUMP_5 = 1    ' GIN 40mL
  DelayMS 2940 '2300
  PUMP_5 = 0
  DelayMS T1

  PUMP_2 = 1    ' pineapple 60mL
  DelayMS 3800 '3900
  PUMP_2 = 0
Case 14
  GoSub SENSOR_1
  GoSub SENSOR_3
  GoSub SENSOR_5
  PUMP_3 = 1    ' VODKA 15mL
  DelayMS 1300
  PUMP_3 = 0
  DelayMS T1

  PUMP_5 = 1    ' GIN 15mL
  DelayMS 1000
  PUMP_5 = 0
  DelayMS T1
```

```
    PUMP_1 = 1    ' ORANGE 70mL
    DelayMS 4550
    PUMP_1 = 0
  Case 15
    GoSub SENSOR_4
    GoSub SENSOR_2
    PUMP_4 = 1    ' tequila 50mL
    DelayMS 3000
    PUMP_4 = 0
    DelayMS T1

    PUMP_2 = 1    ' pineapple 50mL
    DelayMS 3200
    PUMP_2 = 0
  EndSelect

  PORTD = 0
  DelayMS T2
  GoTo SELECTION

SENSOR_1:
  If PORTA.0 = 1 Then
   Cls
   Print At 2, 1, "    NOT AVAILABLE!  "
   DelayMS 2000
   GoTo BEGIN
  EndIf
  Return

SENSOR_2:
  If PORTA.1 = 1 Then
   Cls
   Print At 2, 1, "    NOT AVAILABLE!  "
   DelayMS 2000
   GoTo BEGIN
  EndIf
  Return

SENSOR_3:
  If PORTA.2 = 1 Then
   Cls
   Print At 2, 1, "    NOT AVAILABLE!  "
   DelayMS 2000
```

```
    GoTo BEGIN
   EndIf
   Return

SENSOR_4:
   If PORTA.3 = 1 Then
    Cls
    Print At 2, 1, "    NOT AVAILABLE!  "
    DelayMS 2000
    GoTo BEGIN
   EndIf
   Return

SENSOR_5:
   If PORTA.4 = 1 Then
    Cls
    Print At 2, 1, "    NOT AVAILABLE!  "
    DelayMS 2000
    GoTo BEGIN
   EndIf
   Return


KEYPAD:
   While PORTB.4=0 Or PORTB.5=0 Or PORTB.6=0   ' debouncer
   DelayMS 50
   Wend
   DelayMS 50
getkey0:
   key = 0
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  ROW1
       PORTB = %00001110
       If PORTB.4 = 0 Then
       key = 1
       GoTo exit                          '"1"
       EndIf

       If PORTB.5 = 0 Then
       key = 2
       GoTo exit                          '"2"
       EndIf

       If PORTB.6 = 0 Then
       key = 3
```

```
        GoTo exit                     '"3"
        EndIf
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  ROW2
        PORTB = %00001101
        If PORTB.4 = 0 Then
        key = 4
        GoTo exit                     '"4"
        EndIf

        If PORTB.5 = 0 Then
        key = 5
        GoTo exit                     '"5"
        EndIf

        If PORTB.6 = 0 Then
        key = 6
        GoTo exit                     '"6"
        EndIf
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  ROW3
        PORTB = %00001011
        If PORTB.4 = 0 Then
        key = 7
        GoTo exit                     '"7"
        EndIf

        If PORTB.5 = 0 Then
        key = 8
        GoTo exit                     '"8"
        EndIf

        If PORTB.6 = 0 Then
        key = 9
        GoTo exit                     '"9"
        EndIf
'>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  ROW4
        PORTB = %00000111
        If PORTB.4 = 0 Then
        key = 10
        GoTo exit                     '"*"
        EndIf

        If PORTB.5 = 0 Then
        key = 11
        GoTo exit                     '"0"
```

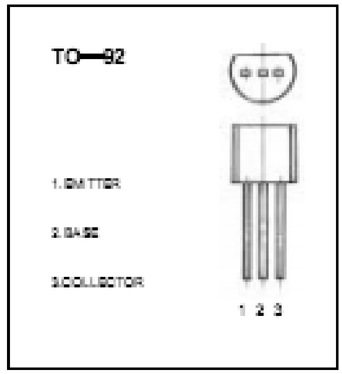
```
        EndIf

        If PORTB.6 = 0 Then
        key = 12
        GoTo exit                          '"#"
        EndIf
        GoTo getkey0
    exit:
    Return

End
```

APPENDIX D
Data Sheets

9018 NPN SILICON TRANSISTOR



FEATURES

Power dissipation
P_{CM} : 0.31 W（Tamb=25℃）
Collector current
I_{CM} : 0.05 A
Collector-base voltage
$V_{(BR)CBO}$: 25 V
Operating and storage junction temperature range
T_J，T_{stg}: -55℃ to +150℃

ELECTRICAL CHARACTERISTICS（Tamb=25℃ unless otherwise specified）

Parameter	Symbol	Test conditions	MIN	TYP	MAX	UNIT
Collector-base breakdown voltage	$V_{(BR)CBO}$	I_C= 100 μA，I_E=0	25			V
Collector-emitter breakdown voltage	$V_{(BR)CEO}$	I_C= 0.1 mA，I_B=0	18			V
Emitter-base breakdown voltage	$V_{(BR)EBO}$	I_E= 100 μA，I_C=0	4			V
Collector cut-off current	I_{CBO}	V_{CB}= 20 V I_E=0			0.1	μA
Collector cut-off current	I_{CEO}	V_{CE}= 15 V I_B=0			0.1	μA
Emitter cut-off current	I_{EBO}	V_{EB}= 3 V，I_C=0			0.1	μA
DC current gain	$H_{FE(1)}$	V_{CE}= 5 V, I_C= 1mA	28		270	
Collector-emitter saturation voltage	V_{CE}(sat)	I_C=10 mA，I_B=1mA			0.5	V
Base-emitter saturation voltage	V_{BE}(sat)	I_C=10mA，I_B=1mA			1.4	V
Transition frequency	f_T	V_{CE}=5 V, I_C=5 mA f =400MHz	600			MHz

CLASSIFICATION OF $H_{FE(1)}$

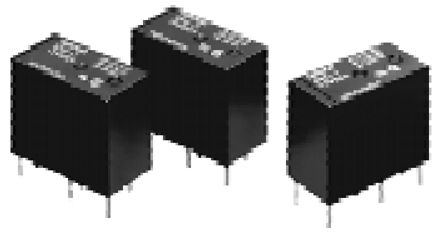
Rank	D	E	F	G	H	I	J
Range	28-45	39-60	54-80	72-108	97-146	132-198	180-270

Wing Shing Computer Components Co., (H.K.)Ltd. Tel:(852)2341 9276 Fax:(852)2797 8153
Homepage: http://www.wingshing.com E-mail: ic@wingshing.com

OMRON®

PCB Relay | G5S

Compact Single-pole Relay for Switching Up To 5 A (Normally Open Contact), Ideal for Fan Control of Air Conditioners, and Heating Control of Small Appliances



- Compact relay with high insulation between coil and contacts
- Up to 5A switching on the NO contacts
- Ensures a withstand impulse voltage of 8,000 V between the coil and contacts
- Class B coil insulation available
- Conforms to UL, CSA, and IEC (TÜV)

Ordering Information

Classification		Enclosure rating	Part number
Single contact	SPDT	Plastic-sealed	G5S-1 (-CB for Class B)
	SPST-NO	Plastic-sealed	G5S-1A (-CB for class B)

Note: When ordering, add the rated coil voltage to the model number.

Example: G5S-1 DC12 — Rated coil voltage

■ MODEL NUMBER LEGEND

G5S-□□-□-□-DC□
1 2 3 4 5

1. Contact Pole
   1: Single pole
2. Contact Form
   nil = 1 form C
   A = 1 form A
3. Insulation class
   nil = standard
   CB = Class B
4. Enclosure
   nil = plastic sealed
   v = vented
5. Rated Coil Voltage
   5, 9, 12, 18, 24, 48 VDC

Specifications

■ COIL RATINGS

Rated voltage	5 VDC	9 VDC	12 VDC	18 VDC	24 VDC	48 VDC
Rated current	80 mA	44.4 mA	33.3 mA	22.2 mA	16.7 mA	8.3 mA
Coil resistance	62.5 Ω	202.5 Ω	360 Ω	810 Ω	1,440 Ω	5760 Ω
Must operate voltage	75% max. of rated voltage					
Must release voltage	5% min. of rated voltage					
Max. voltage	150% of rated voltage at 23°C, 110% of rated voltage at 70°C					
Power consumption	Approx. 400 mW					

Note: Rated current and coil resistance are measured at 23°C with a tolerance of ±10%.

■ CONTACT RATINGS

Load	Resistive load	Inductive load
Rated load	2 A (NO)/2 A (NC) at 277 VAC 5 A (NO)/3 A (NC) at 125 VAC 5 A (NO)/3 A (NC) at 30 VDC	0.5 A at 250 VAC, cosφ=0.4 1A at 250 VAC, cosφ=0.8 0.8 A at 250 VAC, cosφ=0.9
Contact material	Ag	
Rated carry current	5 A (NO)/3 A (NC)	
Max. switching voltage	277 VAC, 30 VDC	
Max. switching current	5 A (NO)/3 A (NC)	1 A
Max. switching capacity	625 VA, 150 W (NO) 375 VA, 90 W (NC)	250 VA
Min. permissible load	10 mA at 5 VDC	

Note: P level: $\lambda_{60}=0.1 \times 10^{-6}$ operation (with an operating frequency of 120 operations/min.)

■ CHARACTERISTICS

Contact resistance (See Note 2.)	100 mΩ max.
Operate time (See Note 3.)	10 ms max.
Release time (See Note 3.)	5 ms max.
Insulation resistance (See Note 4.)	1,000 MΩ min.
Dielectric strength	4000 VAC, 50/60 Hz for 1 min between coil and contacts 750 VAC, 50/60 Hz for 1 min between contacts of same polarity
Impulse withstand voltage	8 kV (1.2 x 50 μs)
Vibration resistance	Destruction: 10 to 55 Hz, 1.5-mm double amplitude for 2 hours Malfunction: 10 to 55 Hz, 1.5-mm double amplitude for 5 minutes
Shock resistance	Destruction: 1,000 m/s^2 (approx. 100G) Malfunction: Energized: 100 m/s^2 (approximately 10G) Non-energized: 50 m/s^2 (approximately 5G)

(This table continues on the next page.)

Characteristics Table - continued from previous page

Life expectancy (See Note 5.)	Mechanical	5,000,000 operations (18,000 operations per hour)
	Electrical	200,000 operations: 1 A (NO)/1 A (NC) at 277-VAC resistive load 3A (NO)/3 A (NC) at 125-VAC resistive load 100,000 operations: 0.8 A (NO)/0.8 A (NC) at 250 VAC, cosφ=0.9 5A (NO)/3 A (NC) at 30-VDC resistive load 50,000 operations: 2 A (NO)/2 A (NC) at 277-VAC resistive load 5A (NO)/3 A (NC) at 125-VAC resistive load
	Switching frequency	1800 operations per hour
Ambient temperature	Operating & storage	–40°C to 70°C (–40°F to 158°F) with no icing or condensation –40°C to 85°C (class B) (–40°F to 185°F)
Ambient humidity	Operating & storage	35% to 85%
Weight		Approx. 8.0 g

Note: 1. The data shown above are initial values.
2. The contact resistance is possible with 1 A applied at 5 VDC using a fall-of-potential method.
3. The operating time is possible with the rated voltage imposed with no contact bounce at an ambient temperature of 23°C.
4. The insulation resistance is possible between coil and contacts and between contacts of the same polarity at 500 VDC.
5. The electrical life data items shown are possible at 23°C.

■ APPROVED STANDARDS

UL508 (File No. E41515)
CSA C22.2 (No. 14) (File No. LR31928)

Model	Coil ratings	Contact ratings	Number of test operations
G5S-1 (-CB) G5S-1A (-CB)	5-48 VDC	0.8 A, 277 VAC (resistive) 0.5 A, 250 VAC (resistive) 2 A, 120 VAC (resistive) 2 A, 30 VDC (resistive) 5 A, 125 VAC (resistive) 1/10 HP, 125 VAC 5 A, 277 VAC (resistive) 1/6 HP, 277 VAC 0.3 A, 110 VDC (resistive) 5 A, 30 VDC (resistive)	6,000

TÜV (IEC 255, VDE0435 File No. R9650783)

Electrical life data are performed at 70°C.

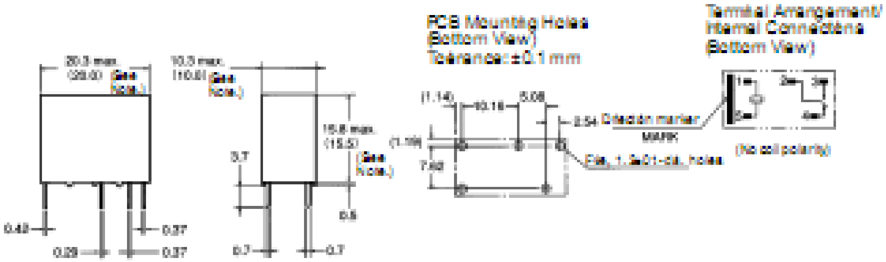
Model	Coil ratings	Contact ratings	Number of test operations
G5S-1 (-CB) G5S-1A (-CB)	5,6,9,12,18,20,24,48	1.5 A, 277 VAC (resistive) 1 A, 250 VAC (resistive) 2 A, 30 VDC (resistive)	30,000 100,000 30,000
		1 A, 250 VAC, cosφ=0.8 0.5 A, 250 VAC, cosφ=0.4 1 A, 250 VAC, cosφ=0.8 (NO only) 1 A, 250 VAC, cosφ=0.8 (NC only)	100,000 30,000 200,000 200,000

Note: Pollution Degree 2, Overvoltage Category II, Material Group III

Dimensions

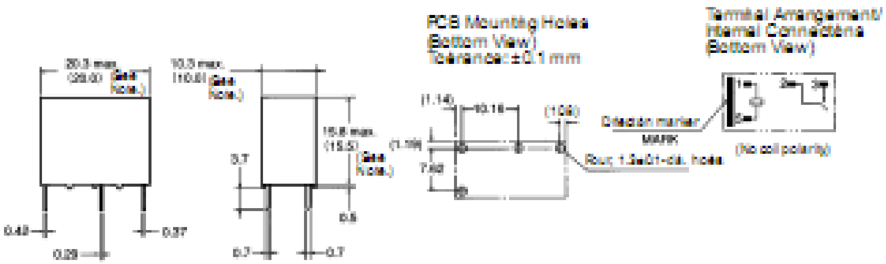
Unit: mm (inch)

■ G5S
SPDT



Note: Values in parentheses are average values.

■ SPST-NO



Note: Values in parentheses are average values.

Precautions

For general precautions on PCB Relays, refer to the precautions provided in *General Information* of the *Relay Product Data Book*.

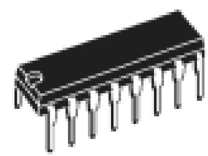
⚠ Caution

Do not touch the terminals of the Relay or the charged part of the socket when power is supplied to the Relay. Otherwise, an electric shock may occur.

ULN2001A-ULN2002A
ULN2003A-ULN2004A

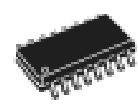
SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A



SO16

ORDERING NUMBERS: ULN2001D2D/3D/4D

DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

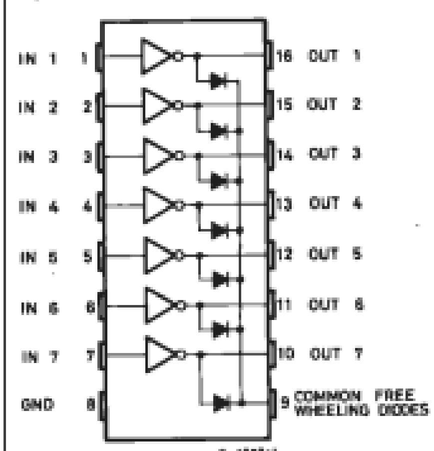
The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6–15V CMOS, PMOS

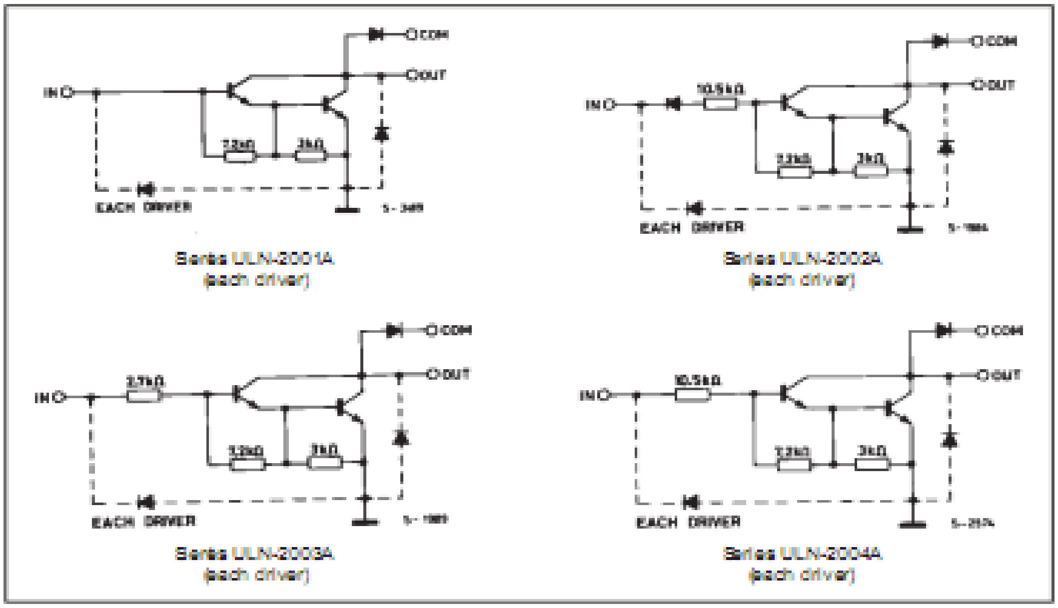
These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal printheads and high power buffers.

The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

PIN CONNECTION



SCHEMATIC DIAGRAM



Series ULN-2001A (each driver)

Series ULN-2002A (each driver)

Series ULN-2003A (each driver)

Series ULN-2004A (each driver)

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter		DIP16	SO16	Unit
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	70	100	°C/W

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V T_{amb} = 70°C, V_{CE} = 50V T_{amb} = 70°C for ULN2002A V_{CE} = 50V, V_i = 6V for ULN2004A V_{CE} = 50V, V_i = 1V			50 100 500 500	µA µA µA µA	1a 1a 1b 1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA I_C = 200 mA, I_B = 350µA I_C = 350mA, I_B = 500µA		0.9 1.1 1.3	1.1 1.3 1.6	V V V	2 2 2
$I_{i(on)}$	Input Current	for ULN2002A, V_i = 17V for ULN2003A, V_i = 3.85V for ULN2004A, V_i = 5V V_i = 12V		0.82 0.93 0.35 1	1.25 1.35 0.5 1.45	mA mA mA mA	3 3 3 3
$I_{i(off)}$	Input Current	T_{amb} = 70°C, I_C = 500µA	50	65		µA	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2V for ULN2002A I_C = 300mA for ULN2003A I_C = 200mA I_C = 250mA I_C = 300mA for ULN2004A I_C = 125mA I_C = 200mA I_C = 275mA I_C = 350mA			 13 2.4 2.7 3 5 6 7 8	V	5
h_{FE}	DC Forward Current Gain	for ULN2001A V_{CE} = 2V, I_C = 350mA	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	
I_R	Clamp Diode Leakage Current	V_R = 50V T_{amb} = 70°C, V_R = 50V			50 100	µA µA	6 6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

TEST CIRCUITS

Figure 1a.

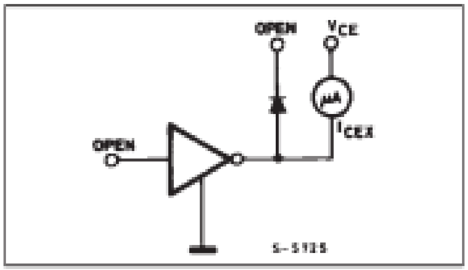


Figure 1b.

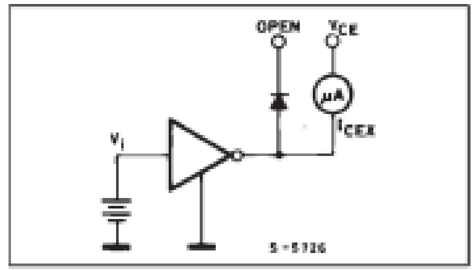


Figure 2.

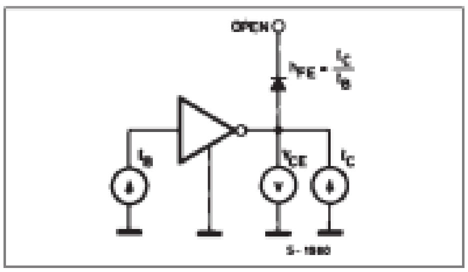


Figure 3.

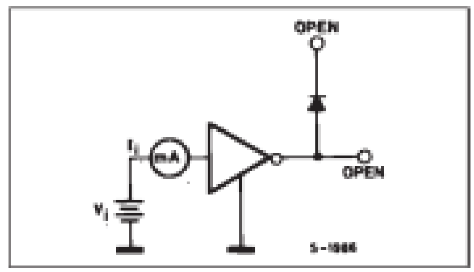


Figure 4.

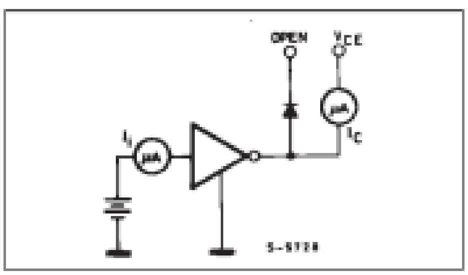


Figure 5.

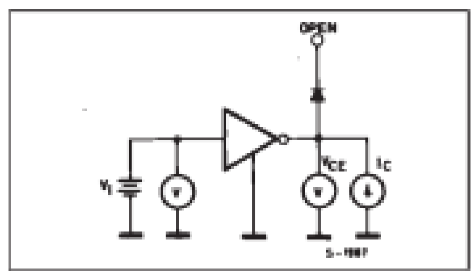


Figure 6.

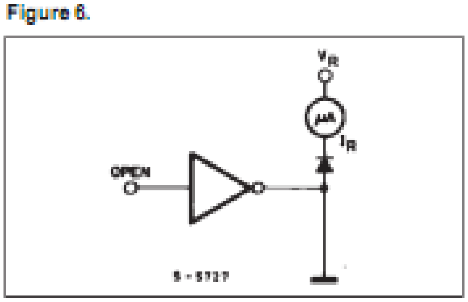
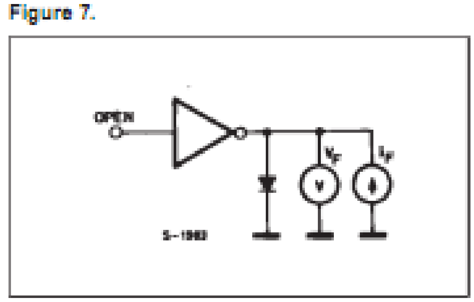


Figure 7.



70

Figure 8: Collector Current versus Input Current

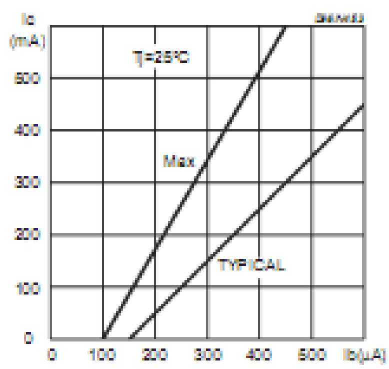


Figure 9: Collector Current versus Saturation Voltage

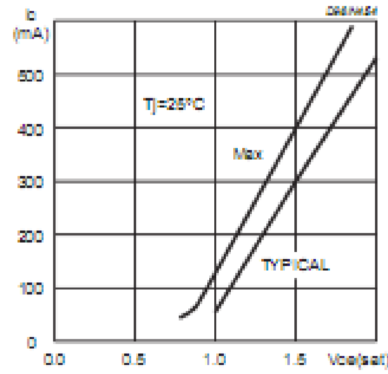


Figure 10: Peak Collector Current versus Duty Cycle

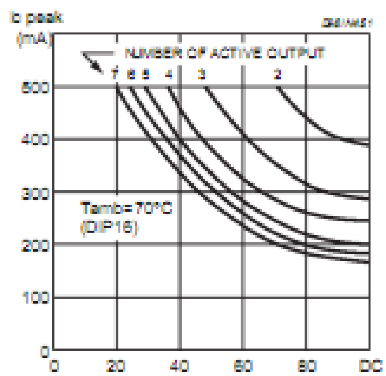
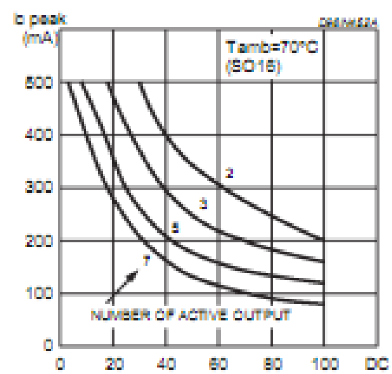
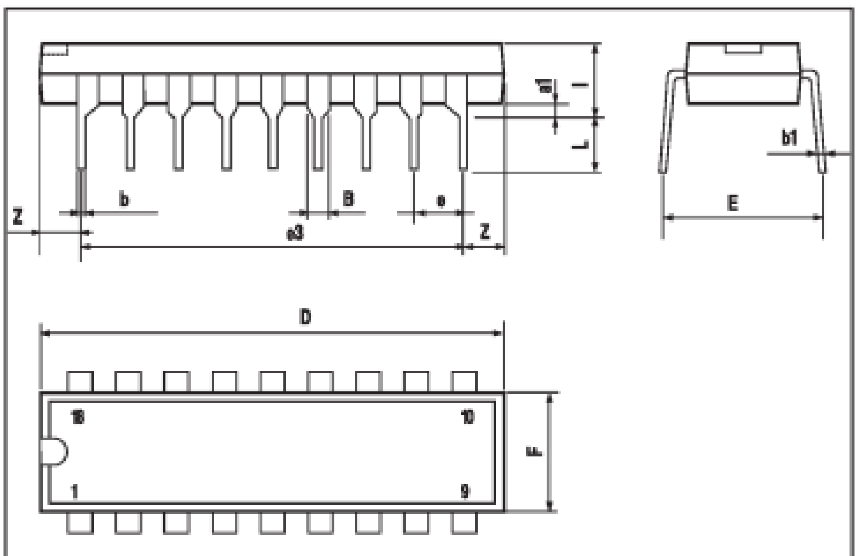


Figure 11: Peak Collector Current versus Duty Cycle



DIP16 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
I			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050



LCD DATASHEET

```
Absolute Maximum Ratings
Power supply for logic (Vdd - Vss)  . . . . 0v min, 6.5v
max
Power supply for LCD drive (Vdd - Vo) . . . 0v min, 6.5v
max

Electrical Characteristics
 Input "high" voltage . . . . . . . . . . . 2.2v min
 Input  "low" voltage . . . . . . . . . . . 0.6v max
Output "high" voltage . . . . . . . . . . . 2.4v min
Output  "low" voltage . . . . . . . . . . . 0.4v max
Power supply current  . . . . . . . . . . . 2.0mA typ, 3.0mA
max
```

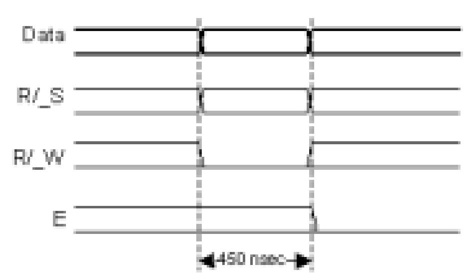
The most common connector used for the 44780 based LCDs is 14 pins in a row, with pin centers 0.100" apart. The pins are wired as:

Pins	Description
1	Ground
2	Vcc
3	Contrast Voltage
4	"R/S" _Instruction/Register Select
5	"R/W" _Read/Write LCD Registers
6	"E" Clock
7 - 14	Data I/O Pins

As you would probably guess from this description, the interface is a parallel bus, allowing simple and fast reading/writing of data to and from the LCD.

This waveform will write an ASCII Byte out to the LCD's screen. The ASCII code to be displayed is eight bits long and is sent to the LCD either four or eight bits at a time. If

LCD Data Write Waveform



four bit mode is used, two "nibbles" of data (Sent high four bits and then low four bits with an "E" Clock pulse with each nibble) are sent to make up a full eight bit transfer. The "E" Clock is used to initiate the data transfer within the LCD.

Sending parallel data as either four or eight bits are the two primary modes of operation. While there are secondary considerations and modes, deciding how to send the data to the LCD is most critical decision to be made for an LCD interface application.

Eight bit mode is best used when speed is required in an application and at least ten I/O pins are available. Four bit mode requires a minimum of six bits. To wire a microcontroller to an LCD in four bit mode, just the top four bits (DB4-7) are written to.

The "R/S" bit is used to select whether data or an instruction is being transferred between the microcontroller and the LCD. If the Bit is set, then the byte at the current LCD "Cursor" Position can be read or written. When the Bit is reset, either an instruction is being sent to the LCD or the execution status of the last instruction is read back (whether or not it has completed).

The different instructions available for use with the 44780 are shown in the table below:

R/S	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Instruction/Description
4	5	14	13	12	11	10	9	8	7	Pins
0	0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	0	1	*	Return Cursor and LCD to Home Position
0	0	0	0	0	0	0	1	ID	S	Set Cursor Move Direction
0	0	0	0	0	0	1	D	C	B	Enable Display/Cursor
0	0	0	0	0	1	SC	RL	*	*	Move Cursor/Shift Display
0	0	0	0	1	DL	N	F	*	*	Set Interface Length
0	0	0	1	A	A	A	A	A	A	Move Cursor into CGRAM
0	0	1	A	A	A	A	A	A	A	Move Cursor to Display
0	1	BF	*	*	*	*	*	*	*	Poll the "Busy Flag"
1	0	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position
1	1	D	D	D	D	D	D	D	D	Read the Character on the Display at the Current Cursor Position

The bit descriptions for the different commands are:

```
"*" - Not Used/Ignored.  This bit can be either "1" or "0"

Set Cursor Move Direction:
  ID - Increment the Cursor After Each Byte Written to Display if Set
   S - Shift Display when Byte Written to Display
```

```
Enable Display/Cursor
   D - Turn Display On(1)/Off(0)
   C - Turn Cursor On(1)/Off(0)
   B - Cursor Blink On(1)/Off(0)

Move Cursor/Shift Display
  SC - Display Shift On(1)/Off(0)
  RL - Direction of Shift Right(1)/Left(0)

Set Interface Length
  DL - Set Data Interface Length 8(1)/4(0)
   N - Number of Display Lines 1(0)/2(1)
   F - Character Font 5x10(1)/5x7(0)

Poll the "Busy Flag"
  BF - This bit is set while the LCD is processing

Move Cursor to CGRAM/Display
   A - Address

Read/Write ASCII to the Display
   D - Data
```

Reading Data back is best used in applications which required data to be moved back and forth on the LCD (such as in applications which scroll data between lines). The "Busy Flag" can be polled to determine when the last instruction that has been sent has completed processing. In most applications, I just tie the "R/W" line to ground because I don't read anything back. This simplifies the application because when data is read back, the microcontroller I/O pins have to be alternated between input and output modes.

For most applications, there really is no reason to read from the LCD. I usually tie "R/W" to ground and just wait the maximum amount of time for each instruction (4.1 msecs for clearing the display or moving the cursor/display to the "home position", 160 usecs for all other commands). As well as making my application software simpler, it also frees up a microcontroller pin for other uses. Different LCDs execute instructions at different rates and to avoid problems later on (such as if the LCD is changed to a slower unit), I recommend just using the maximum delays given above.

In terms of options, I have never seen a 5x10 LCD display. This means that the "F" bit in the "Set Interface Instruction" should always be reset (equal to "0").

Before you can send commands or data to the LCD module, the Module must be initialized. For eight bit mode, this is done using the following series of operations:

1. Wait more than 15 msecs after power is applied.
2. Write 0x030 to LCD and wait 5 msecs for the instruction to complete
3. Write 0x030 to LCD and wait 160 usecs for instruction to complete
4. Write 0x030 AGAIN to LCD and wait 160 usecs or Poll the Busy Flag
5. Set the Operating Characteristics of the LCD

- Write "Set Interface Length"
- Write 0x010 to turn off the Display
- Write 0x001 to Clear the Display
- Write "Set Cursor Move Direction" Setting Cursor Behaviour Bits
- Write "Enable Display/Cursor" & enable Display and Optional Cursor

In describing how the LCD should be initialized in four bit mode, I will specify writing to the LCD in terms of nybbles. This is because initially, just single nybbles are sent (and not two, which make up a byte and a full instruction). As I mentioned above, when a byte is sent, the high nybble is sent before the low nybble and the "E" pin is toggled each time four bits is sent to the LCD. To initialize in four bit mode:

1. Wait more than 15 msecs after power is applied.
2. Write 0x03 to LCD and wait 5 msecs for the instruction to complete
3. Write 0x03 to LCD and wait 160 usecs for instruction to complete
4. Write 0x03 AGAIN to LCD and wait 160 usecs (or poll the Busy Flag)
5. Set the Operating Characteristics of the LCD
   - Write 0x02 to the LCD to Enable Four Bit Mode

     All following instruction/Data Writes require two nybble writes.

   - Write "Set Interface Length"
   - Write 0x01/0x00 to turn off the Display
   - Write 0x00/0x01 to Clear the Display
   - Write "Set Cursor Move Direction" Setting Cursor Behaviour Bits
   - Write "Enable Display/Cursor" & enable Display and Optional Cursor

Once the initialization is complete, the LCD can be written to with data or instructions as required. Each character to display is written like the control bytes, except that the "R/S" line is set. During initializiaion, by setting the "S/C" bit during the "Move Cursor/Shift Display" command, after each character is sent to the LCD, the cursor built into the LCD will increment to the next position (either right or left). Normally, the "S/C" bit is set (equal to "1") along with the "R/L" bit in the "Move Cursor/Shift Display" command for characters to be written from left to right (as with a "Teletype" video display).

One area of confusion is how to move to different locations on the display and, as a follow on, how to move to different lines on an LCD display. The following table shows how different LCD displays that use a single 44780 can be set up with the addresses for specific character locations. The LCDs listed are the most popular arrangements available and the "Layout" is given as number of columns by number of lines:

LCD Layout	Top Left Character	Ninth Character	Second Line	Third Line	Fourth Line	Comments
8x1	0	N/A	N/A	N/A	N/A	Single 44780/No Support Chip

16x1	0	0x040	N/A	N/A	N/A	Single 44780/No Support Chip
16x1	0	8	N/A	N/A	N/A	44780 with Support Chip. This is quite rare
8x2	0	N/A	0x040	N/A	N/A	Single 44780/No Support Chip
10x2	0	8	0x040	N/A	N/A	44780 with Support Chip
16x2	0	8	0x040	N/A	N/A	44780 with Support Chip
20x2	0	8	0x040	N/A	N/A	44780 with Support Chip
24x2	0	8	0x040	N/A	N/A	44780 with Support Chip
30x2	0	8	0x040	N/A	N/A	44780 with Support Chip
32x2	0	8	0x040	N/A	N/A	44780 with Support Chip
40x2	0	8	0x040	N/A	N/A	44780 with Support Chip
16x4	0	8	0x040	0x020	0x060	44780 with Support Chip
20x4	0	8	0x040	0x020	0x060	44780 with Support Chip
40x4	U/N	U/N	U/N	U/N	U/N	Two 44780 with Support Chips. Addressing is device specific

The "Ninth Character" is the position of the Ninth character on the first line.

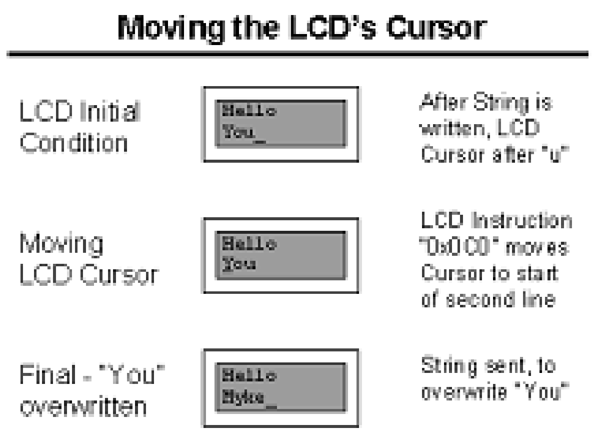
Most LCD displays have a 44780 and support chip to control the operation of the LCD. The 44780 is responsible for the external interface and provides sufficient control lines for sixteen characters on the LCD. The support chip enhances the I/O of the 44780 to support up to 128 characters on an LCD. From the table above, it should be noted that the first two entries ("8x1", "16x1") only have the 44780 and not the support chip. This is why the ninth character in the 16x1 does not "appear" at address 8 and shows up at the address that is common for a two line LCD.

I've included the 40 character by 4 line ("40x4") LCD because it is quite common. Normally, the LCD is wired as two 40x2 displays. The actual connector is normally sixteen bits wide with all the fourteen connections of the 44780 in common, except for

the "E" (Strobe) pins. The "E" strobes are used to address between the areas of the display used by the two devices. The actual pinouts and character addresses for this type of display can vary between manufacturers and display part numbers.

Note that when using any kind of multiple 44780 LCD display, you should probably only display one 44780's Cursor at a time.

Cursors for the 44780 can be turned on as a simple underscore at any time using the "Enable Display/Cursor" LCD instruction and setting the "C" bit. I don't recommend using the "B" ("Block Mode") bit as this causes a flashing full character square to be displayed and it really isn't that attractive.

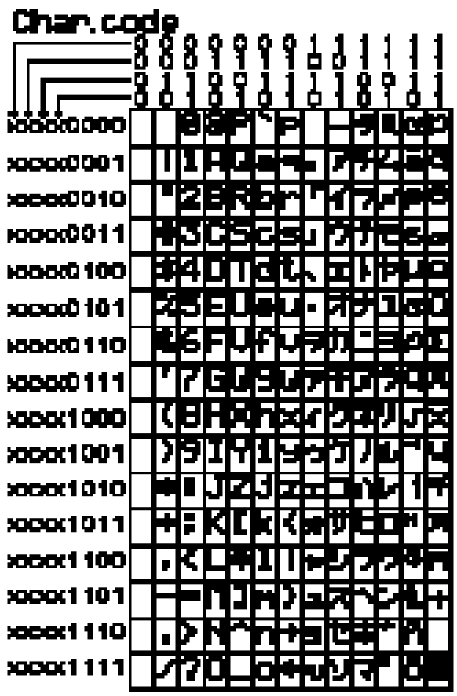


The LCD can be thought of as a "Teletype" display because in normal operation, after a character has been sent to the LCD, the internal "Cursor" is moved one character to the right. The "Clear Display" and "Return Cursor and LCD to Home Position" instructions are used to reset the Cursor's position to the top right character on the display.

To move the Cursor, the "Move Cursor to Display" instruction is used. For this instruction, bit 7 of the instruction byte is set with the remaining seven bits used as the address of the character on the LCD the cursor is to move to. These seven bits provide 128 addresses, which matches the maximum number of LCD character addresses available. The table above should be used to determine the address of a character offset on a particular line of an LCD display.

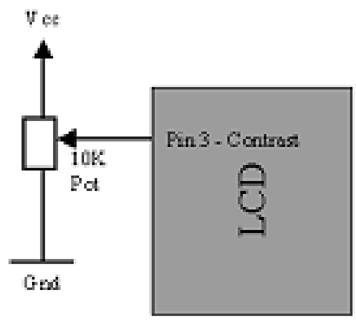
The Character Set available in the 44780 is basically ASCII. I say "basically" because some characters do not follow the ASCII convention fully (probably the most significant difference is 0x05B or "\" is not available). The ASCII Control Characters (0x008 to 0x01F) do not respond as control characters and may display funny (Japanese) characters.

The LCD Character Set shown below is courtesy of Peer Ouwehand and his excellent



character is received. Since all these characters are valid LCD user defined character line definitions, you will find that the SLI-OEM is not interpreting the data correctly. If I was making the "Man" symbol above for displaying on the SLI-OEM, I would use the byte 0x0EE for the first line instead of 0x00E.

LCD Contrast Circuit

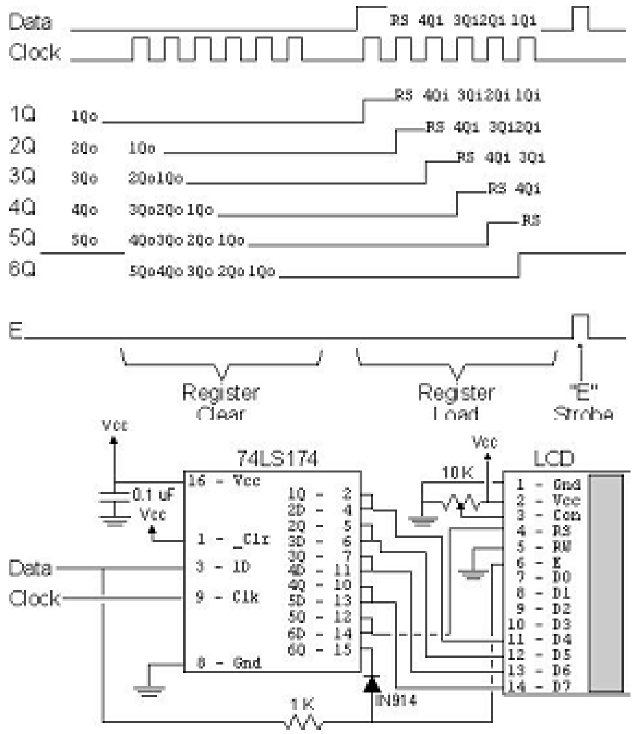


The last aspect of the LCD to discuss is how to specify a contrast voltage to the Display. I typically use a potentiometer wired as a voltage divider. This will provide an easily variable voltage between Ground and Vcc, which will be used to specify the contrast (or "darkness") of the characters on the LCD screen. You may find that different LCDs work differently with lower voltages providing darker characters in some and higher voltages do the same thing in others.

There are a variety of different ways of wiring up an LCD. Above, I noted that the 44780 can interface with four or eight bits. To simplify the demands in microcontrollers, a shift

Eight programmable characters are available and use codes 0x000 to 0x007. They are programmed by pointing the LCD's "Cursor" to the Character

Creating Custom LCD

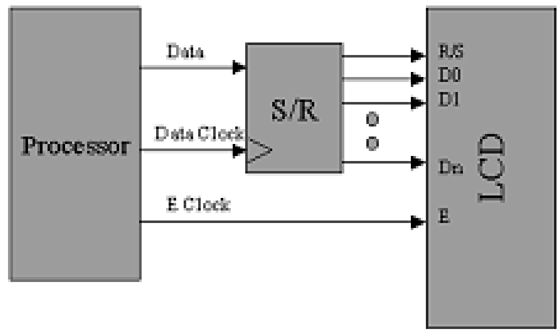


I normally use a 74LS174 wired as a shift register (as is shown in the schematic diagram) instead of a serial-in/parallel-out shift register. This circuit should work without any problems with a dedicated serial-in/parallel-out shift register chip, but the timings/clock polarities may be different. When the 74LS174 is used, note that the data is latched on the rising (from logic "low" to "high") edge of the clock signal.

In the diagram to the right, I have shown how the shift register is written to for this circuit to work. Before data can be written to it, the shift register is cleared by loading every latch with zeros. Next, a "1" (to provide the "E" Gate) is written followed by the "R/S" bit and the four data bits. Once the Data is loaded in correctly, the "Data" line is pulsed to Strobe the "E" bit. The biggest difference between the three wires and two wire interfaces is that the shift register has to be cleared before it can be loaded and the two wire operation requires more than twice the number of clock cycles to load four bits into the LCD. Note: Make sure that before the "E" line is output on "0", there is at least a 450 nsecs delay with no lines changing state.

register is often used (as is shown in the diagram below) to reduce the number of I/O pins

Shift Register LCD Data Write



to three.

This can be further reduced by using the circuit shown below in which the serial data is combined with the contents of the shift register to produce the "E" strobe at the appropriate interval.

This circuit "ANDs" (using the 1K resistor and IN914 diode) the output of the sixth "D-Flip Flop" of the 74LS174 and the "Data" bit from the device writing to the LCD to form the "E" Strobe. This method requires one less pin than the three wire interface and a few more instructions of code.

PIC16F87X
Data Sheet

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

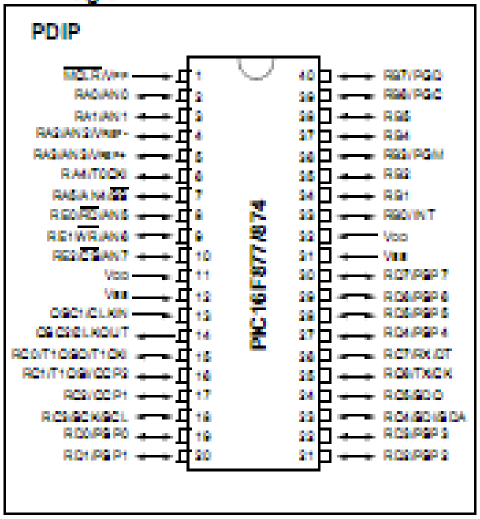
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

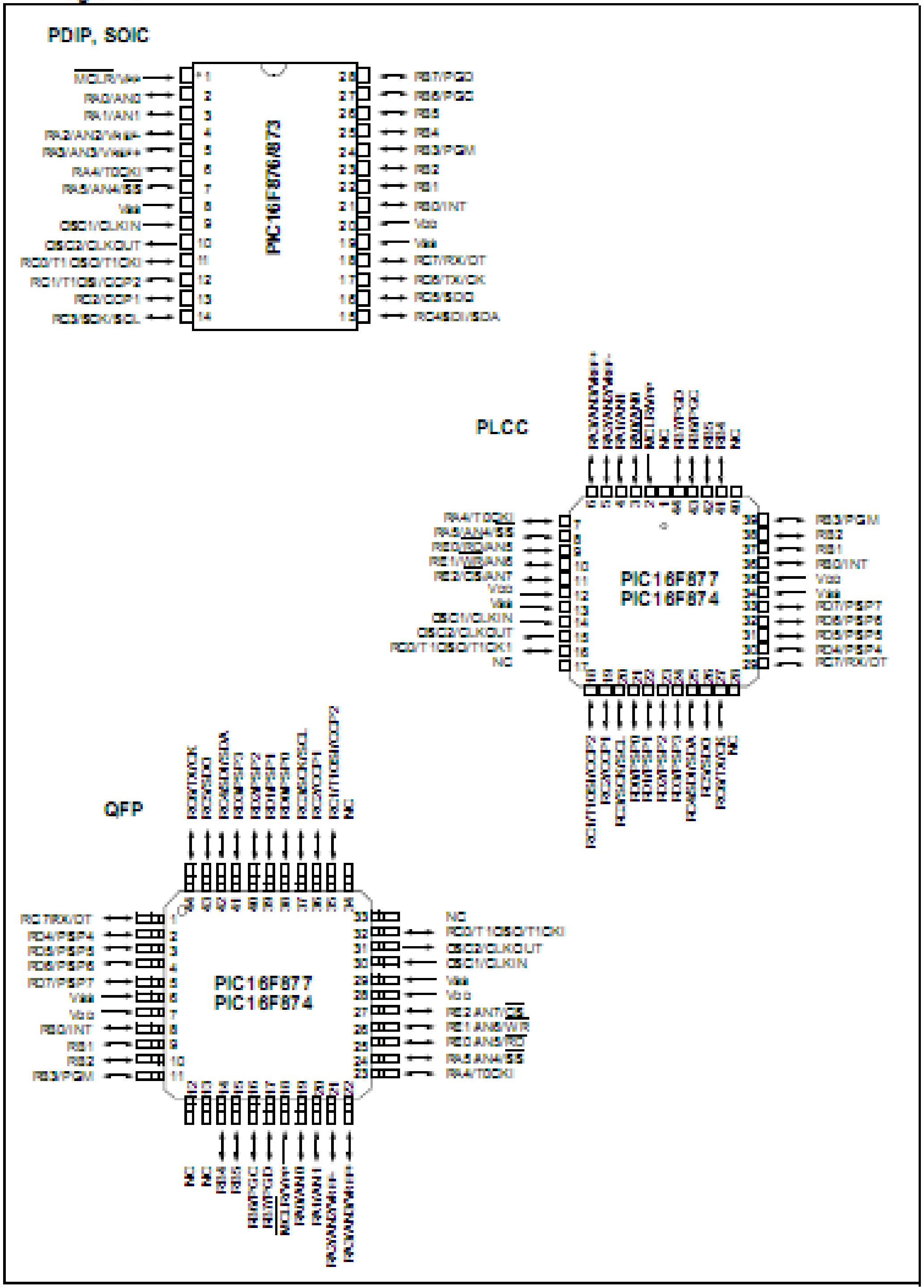
Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Pin Diagrams



Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

1.0 DEVICE OVERVIEW

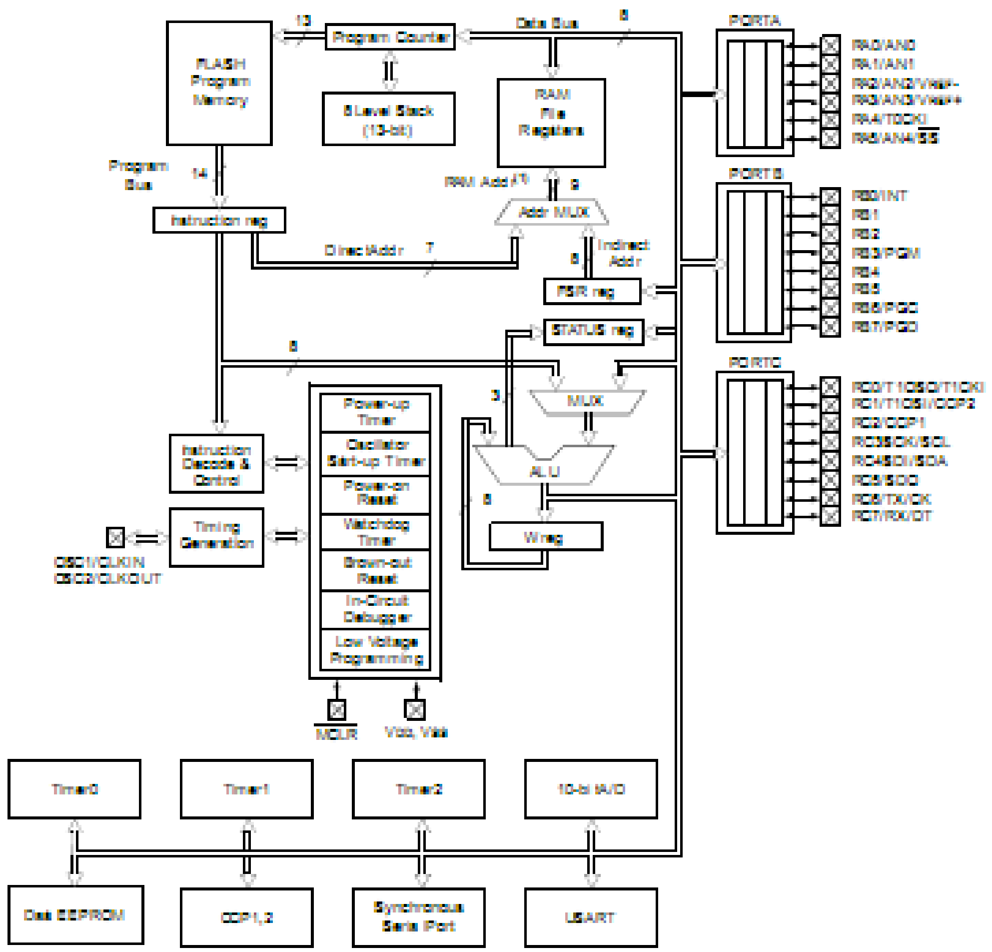
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.

The following device block diagrams are sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

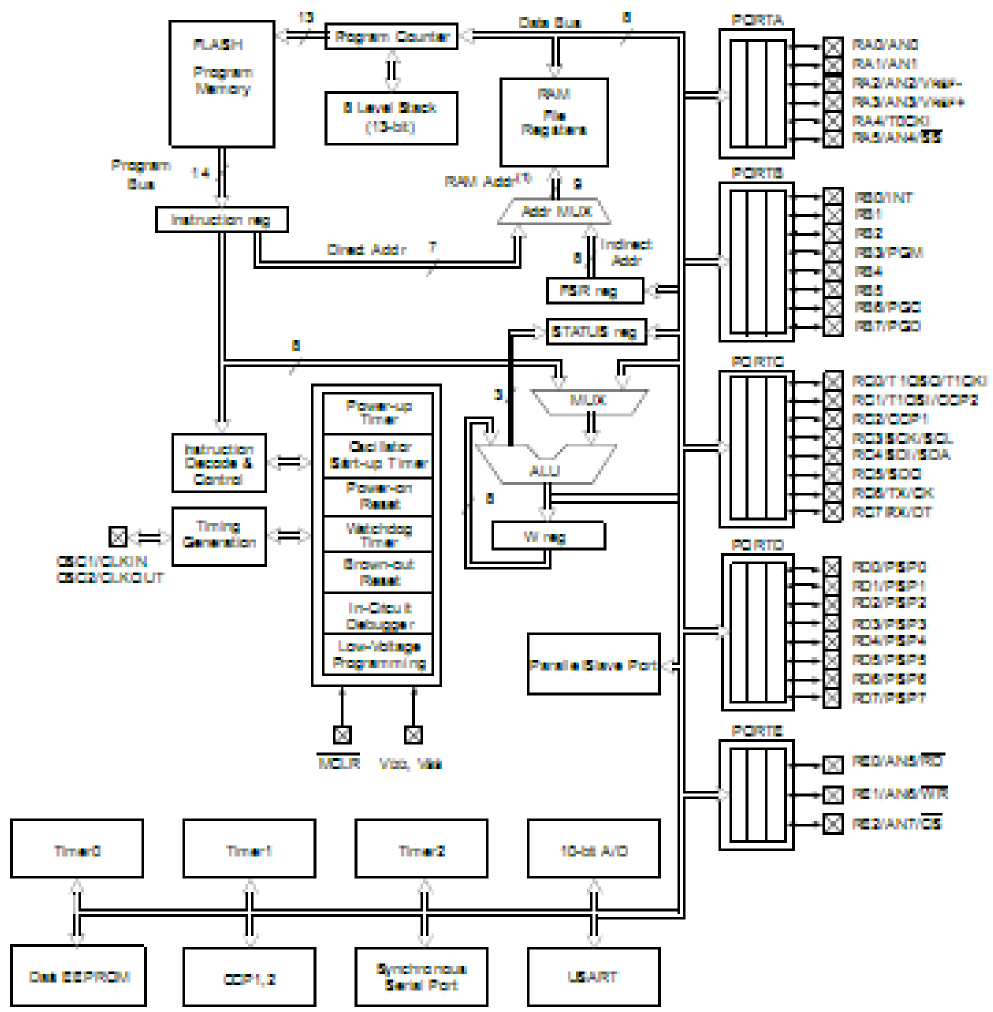
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/V_{PP}	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/V_{REF-}	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/V_{REF+}	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	25	25	I/O	TTL	Interrupt-on-change pin.
RB5	26	26	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
V_{SS}	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
V_{DD}	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS[4]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST[1]	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.
RB5	38	42	15	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	39	43	16	I/O	TTL/ST[2]	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST[2]	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F874 AND PIC16F877 PINOUT DESCRIPTION (CONTINUED)

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I²C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I²C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL(3)	
RD1/PSP1	20	22	39	I/O	ST/TTL(3)	
RD2/PSP2	21	23	40	I/O	ST/TTL(3)	
RD3/PSP3	22	24	41	I/O	ST/TTL(3)	
RD4/PSP4	27	30	2	I/O	ST/TTL(3)	
RD5/PSP5	28	31	3	I/O	ST/TTL(3)	
RD6/PSP6	29	32	4	I/O	ST/TTL(3)	
RD7/PSP7	30	33	5	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	RE0 can also be read control for the parallel slave port, or analog input5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL(3)	RE2 can also be select control for the parallel slave port, or analog input7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.